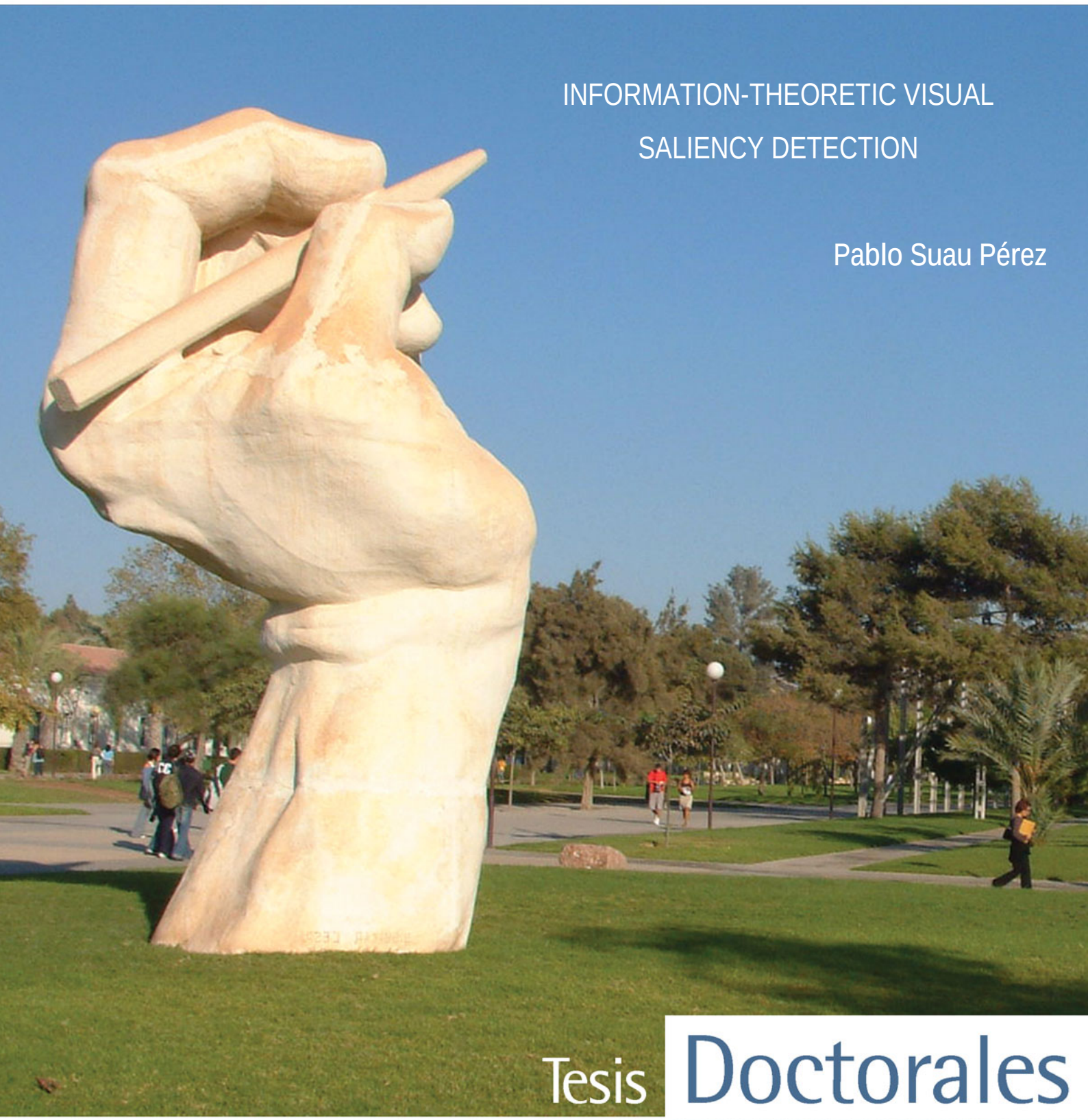
Universitat d'Alacant
Universidad de Alicante

INFORMATION-THEORETIC VISUAL SALIENCY DETECTION

Pablo Suau Pérez



Tesis Doctorales

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE

PH.D. THESIS

INFORMATION-THEORETIC VISUAL SALIENCY DETECTION

Pablo Suau

Supervised by **Dr. Francisco Escolano**

Dept. of Computer Science and Artificial Intelligence
UNIVERSITY OF ALICANTE

June, 2010

Contents

List of Figures

List of Tables

Acknowledgements

It is hard to believe that the moment of writing the acknowledgements section of my thesis has arrived. Looking back to five years ago, I never have thought that eventually I would be able to finish this huge amount of work. Now I am sitting here remembering all the people that helped me to make it happen. Without all of you I would not be so proud of my work today.

It all started exactly five years ago. All my new fellows from the Robot Vision Group, here at the University of Alicante, were very welcoming and they made me feel again excited about my research. I regained the faith in my work and in myself. What they did for me was invaluable: not only I learned more than I did before, but also they made me feel as part of a team of extraordinary individuals. At last I had the work environment I always wished.

Although I am very grateful to all of them, I would like to focus on some of these individuals. Firstly, I would like to give thanks to Francisco Escolano, my supervisor and the person who started it all. His advice helped me to transform my ideas into what my thesis is today. From the beginning, his commitment to my learning and to my goals was remarkable. From him I learned to be a better researcher, to use the best information sources about Computer Vision, to focus on the important, to increase the quality of my work and my experiments, and of course to improve the theoretical background of my methods.

I would also like give thanks to Juan Manuel Sáez for his friendly advice and help throughout my entire career at the University. I remember with joy the first time I entered his office and I asked to him: ”I want to be an University lecturer and to research in Artificial Intelligence. What should I do?”. Several years passed by and he eventually became my fellow, a good

one. He always gave me his helping hand, no matter what I was worried about. I know that I can always count on him. I hope that he can also find support from me whenever he needs it.

My life would have been more boring without the people in my laboratory. Boyán and Jose Manuel hugely helped me. With their constant jokes (some of them, by the way, quite practical) they helped me to learn more about myself. They helped me to develop my ability to laugh at myself, and I think that they contributed to my personal development. They influenced my way of seeing the world today. By the other hand, how to forget all those "deadline parties" that Boyán, Francisco and I had just before the deadline of several conferences? They were hard, but they also were funny.

Other people who always responded to my requests were Miguel Ángel Lozano and Antonio Peñalver. Thank you for your help with my experiments, for your ideas and four your kindness.

But the greatest support during the last years did not come from the University or the laboratory. It came from the person I love most in this world. Beatriz, you are the best traveling companion I can imagine. You have been always there. You comforted me in the bad times, when I was considering to give up. You celebrated my victories and my joys. When you first met me, seven years ago, I was just starting my thesis, so you have always wondered how my life would be if I finished it. We will now find out together. Wacken 2010 will be our most intense experience ever!

Finally, I would like to express my gratitude to my family. Thank you four your support and understanding. You know that I am always busy, but you are always present in my mind. Thank you mother, father, Tono, Mamen and also thank you, grandparents. You made it easier.

Agradecimientos

Casi no puedo creer que haya llegado el momento de escribir la sección de agradecimientos de mi tesis. Hace cinco años no hubiera imaginado que sería capaz de finalizar este proyecto. Y la verdad es que ahora mismo no puedo evitar pensar en toda la gente que me ayudó a conseguirlo. Sin todos vosotros, hoy en día no estaría tan orgulloso de mi trabajo.

Todo empezó hace exactamente cinco años. Todos mis compañeros del Robot Vision Group, aquí en la universidad de Alicante, me recibieron con los brazos abiertos y me hicieron sentirme de nuevo entusiasmado con mi investigación. Recuperé la fe en mi trabajo y en mí mismo. Lo que hicieron por mí no tiene precio: no sólo aprendí más de lo que había aprendido hasta el momento, sino que me hicieron sentir que formaba parte de un equipo de personas extraordinarias. Por fin tenía el ambiente de trabajo que siempre había deseado.

Aunque estoy muy agradecido a todos ellos, me gustaría centrarme en algunas de estas personas. Me gustaría empezar dándole gracias a Francisco Escolano, mi tutor. Gracias a él todo esto empezó. Sus consejos ayudaron a convertir mis ideas en lo que mi tesis es hoy en día. Desde el principio, su dedicación a mi proceso de aprendizaje y a mis metas fue impresionante. Gracias a él aprendí a ser un mejor investigador, a utilizar las mejores fuentes de información sobre Visión Artificial, a centrarme en lo importante, a mejorar la calidad de mi trabajo y experimentos, y por supuesto a mejorar el trasfondo teórico de mis métodos.

Me gustaría también agradecerle a Juan Manuel Sáez su ayuda y sus consejos a lo largo de toda mi carrera en la universidad. Sonrío al recordar la primera vez que entré en su despacho y le pregunté: "Quiero ser profesor en la universidad e investigar en Inteligencia Artificial, ¿qué es lo que tengo

que hacer?". Tras unos cuantos años ha pasado a ser mi compañero, y uno muy bueno. Siempre me ha ofrecido su ayuda, sin importar qué necesitara. Sé que siempre puedo contar con el. Espero que el también encuentre apoyo en mi siempre que lo necesite.

Mi vida hubiera sido mucho más aburrida sin la gente de mi laboratorio. Boyán y Jose Manuel me ayudaron mucho. Gracias a sus bromas constantes (algunas de ellas bastante pesadas) me ayudaron a aprender más sobre mi mismo. Gracias a ellos aprendí a reírme de mi mismo, y pienso que han aportado bastante a mi desarrollo como persona. Han influido mucho en mi forma de ver las cosas. Por otra parte, ¿cómo olvidar esas "deadline parties" que Boyán, Francisco y yo organizábamos justo antes de la fecha límite de entrega de trabajos de algunas conferencias? Eran momentos duros, pero también muy divertidos.

Otras personas que también han respondido siempre a mis peticiones han sido Miguel Ángel y Antonio Peñalver. Gracias por ayudarme con mis experimentos y mis dudas, por vuestras ideas y por vuestra amabilidad.

Pero sin duda el mayor apoyo no lo he recibido de ningún compañero de la universidad o del laboratorio, sino que de la persona a la que más quiero en este mundo. Beatriz, eres la mejor compañera de viaje que puedo imaginar. Siempre has estado ahí. Me hiciste sentir mejor en los momentos difíciles, incluso cuando tuve la tentación de dejarlo todo. Has celebrado mis éxitos y mis alegrías. Cuando me conociste por primera vez, hace siete años, todavía estaba empezando mi doctorado, así que siempre te has preguntado cómo sería mi vida si no tuviera que preocuparme tanto de la tesis. Vamos a descubrirlo juntos. ¡El Wacken 2010 va a ser nuestra mejor experiencia!

Finalmente, me gustaría expresar mi gratitud hacia mi familia. Os doy las gracias por vuestro apoyo y comprensión. Sé que siempre estoy ocupado, pero siempre estáis presentes en mi vida. Gracias mamá, papá, Tono, Mamen, y también gracias a vosotros, Abu y Lola. Gracias a vosotros ha sido todo más fácil.

Abstract

In this thesis we present two improvements of the Scale Saliency algorithm proposed by T. Kadir and M. Brady. The Scale Saliency algorithm extracts regions of interest from images. These regions can be used in high-level vision applications. Since it is based on Information Theory, it is theoretically sound: extracted regions are those corresponding to unpredictable or the most informative events. However, in the current state of the art in feature extraction algorithms, the Scale Saliency algorithm is the less computationally efficient method.

Firstly, we address the problem of computational efficiency. We propose a filter that discards points of an image prior to the application of the Scale Saliency algorithm. This filtering process remarkably decreases the computation time of the Kadir and Brady method, with a low error rate. The method uses Bayesian inference in order to learn a saliency threshold for a set of images. Then, using this threshold and Information Theory, a decision rule is defined. The aim of this rule is to discard points of the image that probably do not belong to the most salient regions of that image.

Secondly, we propose a method to decrease the computational order of the algorithm with respect to data dimensionality. The Scale Saliency algorithm is usually applied to grayscale images, but it can also be easily applied to higher-dimensional information, like RGB color images, due to how the Information Theory measures used during the algorithm are estimated. However, the complexity of the algorithm exponentially increases with data dimensionality.

We assess different entropy and divergence estimation methods, based on graphs and data partition, in order to design a multi-dimensional version of the Scale Saliency algorithm. We also propose a new divergence measure

based on one of these estimators. Not only the complexity order of our multi-dimensional Scale Saliency algorithm decreases from exponential to linear (with respect to data dimensionality), but also it is computationally efficient. It is able to process up to 31D data in a few minutes (the Kadir and Brady algorithm requires several hours to process a 4D image).

We present, in both cases, an example of application to a real-world problem. We apply our Bayesian filtering of the Scale Saliency algorithm to the robot localization problem. In the case of multi-dimensional Scale Saliency, it is applied to the texture categorization problem.

Resumen

En esta tesis presentamos dos modificaciones del algoritmo Scale Saliency creado por T. Kadir y M. Brady. Se trata de un algoritmo de extracción de regiones de interés en imágenes, de tal forma que éstas puedan ser utilizadas en aplicaciones de visión de alto nivel. El algoritmo Scale Saliency se apoya en un trasfondo teórico sólido, ya que se basa en la Teoría de la Información: las regiones extraídas serán aquellas que se correspondan con eventos impredecibles o que provean la máxima información posible. Sin embargo, dentro del estado del arte en el campo de los algoritmos de extracción de características, este algoritmo es el menos eficiente temporalmente.

En primer lugar tratamos el problema de la eficiencia temporal proponiendo un filtro que permite descartar puntos de la imagen antes de la aplicación del algoritmo Scale Saliency. Este proceso de filtrado disminuye notablemente el tiempo de ejecución del algoritmo de Kadir y Brady, con una baja tasa de error. El método se basa en el uso de inferencia Bayesiana para el aprendizaje de un umbral de saliencia válido para un conjunto de imágenes. Gracias a este umbral se puede definir una regla de decisión por medio de la Teoría de la Información para descartar los puntos de la imagen que probablemente no forman parte de las regiones más salientes de la imagen.

En segundo lugar proponemos un algoritmo para disminuir la complejidad del algoritmo con respecto a la dimensionalidad de los datos. El algoritmo Scale Saliency es usado normalmente con imágenes en tonos de gris, pero debido a cómo estima las medidas relacionadas con la Teoría del a Información puede también ser fácilmente aplicado a datos de mayor dimensionalidad, como imágenes en color. Sin embargo, la complejidad del algoritmo crece exponencialmente con respecto a la dimensionalidad de los datos.

Para conseguir esto estudiamos diferentes métodos de estimación, basados tanto en grafos como en partición de datos, para diseñar una versión multidimensional del algoritmo Scale Saliency. También proponemos una nueva medida de divergencia basada en uno de estos algoritmos de estimación. Conseguimos no sólo disminuir la complejidad del algoritmo de exponencial a lineal, sino también que sea lo suficientemente eficiente como para procesar datos de hasta 31 dimensiones en unos pocos minutos (el algoritmo de Kadir y Brady necesitaría varias horas para procesar una imagen compuesta de datos en 4 dimensiones).

En el caso de ambas aportaciones presentamos un ejemplo de aplicación. Nuestro filtro Bayesiano es aplicado al problema de la localización robótica. Aplicamos también nuestra versión multidimensional del algoritmo Scale Saliency al problema de la categorización de texturas.

Chapter 1

Motivation and goals

1.1 Motivation

Computer Vision may be defined as the field of Computer Science whose aim is the design of systems or machines that are able to interpret what they *see* through any image sensor, like a conventional camera. It is a young discipline: the first feasible Computer Vision systems were reported back in the seventies. Due to this fact, Computer Vision is an exciting field: new ideas are constantly being generated, and also new applications are constantly being devised, like industrial quality control, medical image analysis or robot localization and mapping. Several applications are even reaching the consumer market: cameras with face detection, intelligent image search and retrieval systems, tracking in surveillance systems, and so on. However, Computer Vision is still an immature field. Computer Vision algorithms tend to focus on solving a very specific problem, and the generalization of these algorithms is difficult or impossible. In this regard, the inclusion of Machine Learning algorithms in Computer Vision systems provide a new approach to adaptative systems. Biological vision is another important source of inspiration for several Computer Vision scientists, who devise algorithms to model components from human or animal vision systems.

High-level Computer Vision applications usually rely on low-level information provided by image processing or feature extraction algorithms. For instance, edges can be treated as visual clues in the task of searching for a given object in an image. Furthermore, image features provide a

sparse representation of the image; high-level vision tasks do not need to process the complete image. They can focus on these features instead, saving computation time. It is clear that the performance of high-level vision tasks strongly depends on the type and the quality of the extracted low-level features. Thus, feature extraction algorithms are a key part of Computer Vision systems. Feature extraction algorithms are aimed to extract visual features that satisfy desirable properties, like being informative, distinguishable, and invariant to image transformations, like variation in scale, lighting conditions or viewpoint. Several authors prefer the term *covariant* referring to image features that adapt to the transformation applied to the image.

Computer Vision is related to different fields, like Artificial Intelligence, Physics, Machine Learning or Mathematics. Information Theory, a branch of Mathematics that studies information quantification, has been widely applied to Computer Vision in the past. The Scale Saliency algorithm is an example of information-theoretic algorithm. It is a feature extraction algorithm, that searches for salient or high-informative regions on images. It is an interesting algorithm, not only because it was reported to yield good features for the task of image categorization, but also because it is theoretically sound: it uses Information Theory in order to locate the most informative image features. However, its time complexity is so high when compared to the rest of state-of-the-art invariant feature extraction algorithms. This fact restricts the use of the algorithm to systems that do not operate in real time.

1.2 Goals

Our primary goal is to increase the efficiency of the Scale Saliency algorithm. The main step of the algorithm involves the estimation of entropy for all pixels in a range of scales. Each scale defines the area around the pixel from which entropy is estimated. The main bottleneck of the algorithm is precisely the computation of entropy in the complete range of scales. Our hypothesis is that only the estimation at a given scale is required in order to mark a subset of pixels on the image as *non-interesting*. Then, the complete

algorithm would be only applied to the rest of pixels, saving computation time.

Which scale should we process in order to discard pixels prior to the application of the Scale Saliency algorithm? We start from an intuitive idea: if a large region is homogeneous (and, as a consequence, not informative), then subparts of that region, at lower scales, will also be homogeneous (and not informative). Thus, the estimation of entropy at the highest scale may help in the task of detecting these non-informative pixels. In order to support this idea, we need to analyze the evolution of the entropy function in the scale-space. The aim of this analysis is to formalize the relationship between entropy at higher and lower scales.

Once this relationship has been established, it is necessary to define a decision rule that marks non-interesting pixels, so these pixels can be filtered before the application of the Scale Saliency algorithm. This decision rule should be general or, at least, it should be applicable to a set of images. Therefore, we will propose a classification algorithm that, given the entropy of a pixel at the highest scale, classifies it as suitable or non-suitable for further processing. Our goal here is to apply Information Theory and Bayesian inference in order to implement this minimal risk classification algorithm, since both have been previously and successfully applied to similar problems.

Another goal of this thesis, related with the primary one, is to study of the application of the Scale Saliency algorithm to high-dimensional data. Scale Saliency is usually applied to graylevel intensity images, for which entropy is estimated by means of intensity frequency histograms. It can easily be applied to color images by increasing the histogram dimensionality, or even to higher dimensional data. However, the computational complexity of the algorithm increases exponentially with data dimensionality, due precisely to the histogram based estimation. Furthermore, high-dimensional data yields sparser histograms that are less informative.

We can find alternative entropy estimation approaches based on Minimal Spanning Trees, K-Nearest Neighbor Graphs and other types of data partition in the literature. We need to evaluate the suitability of these kind of methods to the entropy estimation step of the Scale Saliency algorithm. Another step in the Scale Saliency algorithm that depends on histogram estimation

is the weighting of entropy peaks. In this step the algorithm estimates self-dissimilarity between scales in order to penalize those features that are salient in a wide range of scales. We will evaluate graphs and data partition based divergence measures to totally discard the use of histograms during the Scale Saliency algorithm.

An additional (and also important) goal of our thesis is to apply our contributions to real-world problems. We will need to find applications for which our contributions to the Scale Saliency algorithm actually provide clear improvement of efficiency and/or performance.

1.3 Thesis overview

After introducing the motivations and goals of this thesis in Chapter 1, we present in Chapter 2 a detailed survey of the state of the art in the field of visual feature extraction methods. This survey is focused on interest point and interest region detection algorithms. This chapter also defines the term *local visual saliency* and it summarizes the Scale Saliency algorithm.

In Chapter 3 we present a detailed description of our Bayesian filtering approach, aimed to build a decision rule to discard non-interesting pixels before applying the Scale Saliency algorithm to an image. This approach emerges from an analysis of the evolution of the entropy function in the scale-space. Our experiments demonstrate the effect of different parameters on the results of our algorithm. We also show how to apply our algorithm to the robot localization problem.

Next, in Chapter 4, we focus on the design of a multi-dimensional Scale Saliency algorithm. We survey several entropy and divergence estimators based on entropic graphs and data partition. The experimental results compare our multi-dimensional method to the Scale Saliency algorithm in terms of number and quality of extracted features and computation time. We apply our multi-dimensional Scale Saliency algorithm to the texture categorization problem.

In Chapter 5 we draw the main conclusions derived from our work in this thesis. We also discuss several aspects of this work that may be subject to further analysis and research.

Finally, in Annex A we present a list of our publications in journals and conferences related to this thesis, and in Annex B we include a Spanish summary of the thesis.

1.4 Main contributions

These are the main contributions of our thesis:

- Firstly, in Chapter 2 we present a complete survey of the evolution of visual feature extraction algorithms and the state of the art in this field. Although we can find several surveys in the literature, we are not aware of any other one that covers the complete history of this kind of algorithms spanning from the seventies to the present day. Furthermore, we also summarize recent algorithms that were not included in previous surveys (see Fig. 1.1).

- Secondly, in Chapter 3 we present a method aimed to remarkably increase the computational efficiency of the Scale Saliency algorithm. First of all, we analyze the evolution of the entropy function in the scale-space. This analysis is the basis of a first filtering method, that discards non-interesting points before applying the Scale Saliency algorithm. This approach is limited; it does not provide a priori information to select a valid entropy threshold. We extend this first method by means of Bayesian inference and Information Theory in order to learn a threshold applicable to a set of images. Information Theory also provides an evaluation tool, Chernoff Information, that assess the applicability of our algorithm to a given set of images (see Fig. 1.2).

- Thirdly, in Chapter 4 we modify the Scale Saliency algorithm, achieving a remarkably decrease in computational complexity with respect to data dimensionality (from exponential to linear). We study different entropy and divergence estimation methods whose complexity does not strongly depend on data dimensionality, in order to avoid the use of histogram based estimations in the Scale Saliency algorithm. These methods are based on graphs and data partition. We also

introduce a new divergence measure based on data partition. We experimentally compare these approaches in order to propose a feasible multi-dimensional Scale Saliency algorithm (see Fig. 1.3).

- Finally, we apply our algorithms to real-world problems. In Chapter 3 we see how to apply our filtering algorithm to the robot localization problem. Regarding multi-dimensional Scale Saliency, in Chapter 4 it is applied to the texture categorization problem.

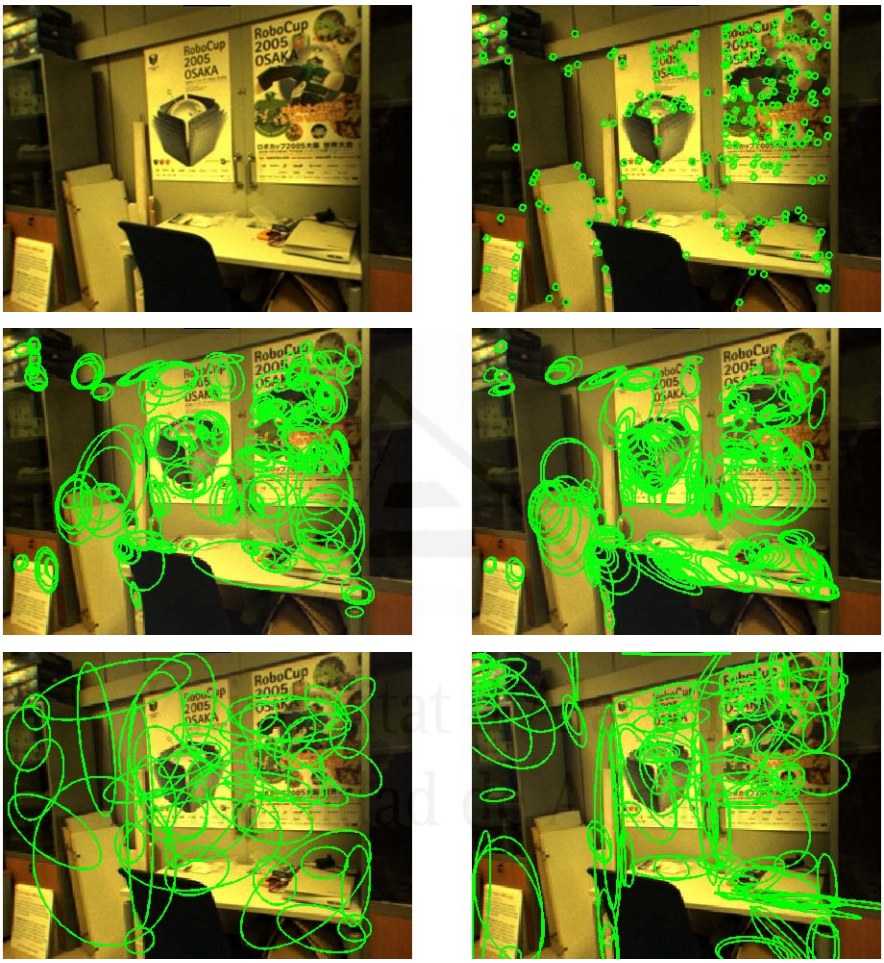


Figure 1.1: Example of application of several image feature extraction algorithms. From left to right and from top to bottom: input image, Harris corner detector, Harris affine, Hessian affine, Intensity Based Regions, Maximally Stable Extremal Regions.

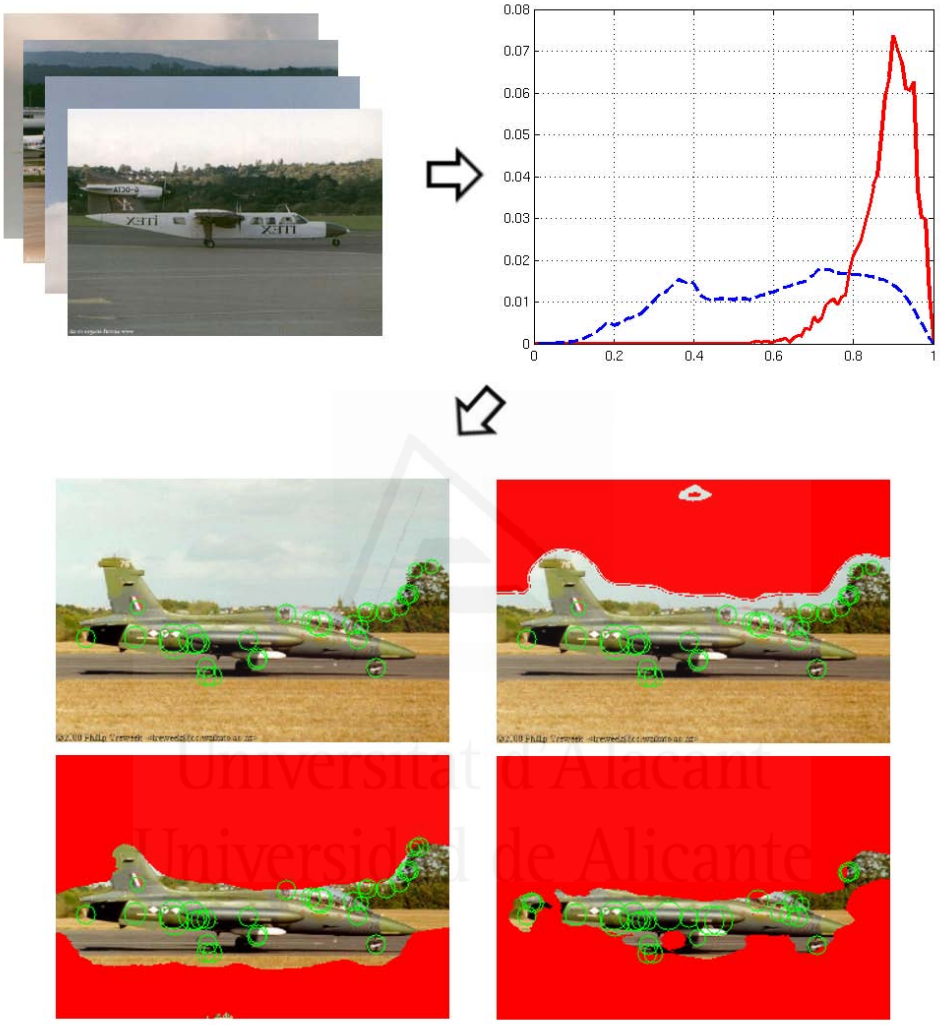


Figure 1.2: Example of application of our Bayesian filtering method. From a set of images belonging to the same image category (top left) we build a log-likelihood based classifier (top right). This classifier discards non-salient points before applying Scale Saliency. Information Theory provides a range of valid threshold values (bottom).

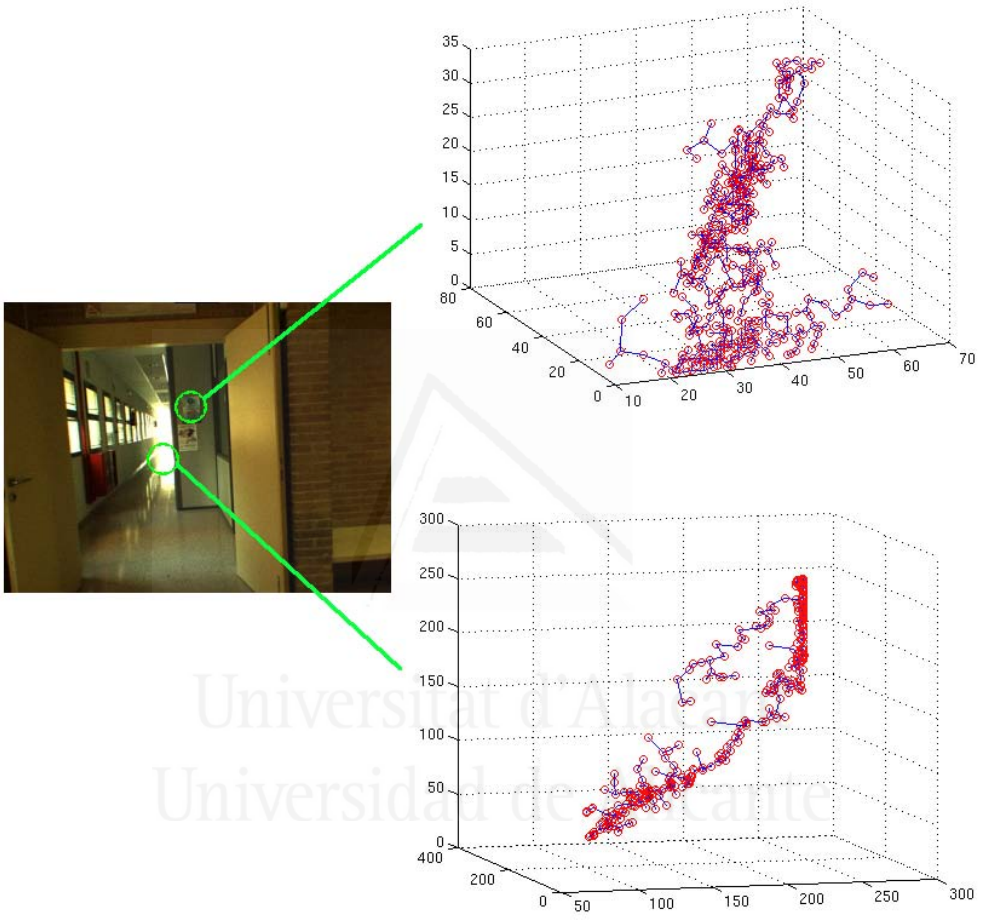


Figure 1.3: Example of visual saliency estimation from Minimal Spanning Trees. Each 3D node represents the RGB values of a pixel in any of the two selected image regions. The upper image region is more salient (the total length of the tree is 894.11) than the lower one (in this case, the total length of the tree is 575.03).

Chapter 2

Related work and theoretical background

In this chapter we present the context in which our thesis is conducted and several concepts that will be used throughout the rest of chapters. As stated in the Introduction, the aim of our work is to improve the efficiency of the Kadir and Brady Scale Saliency algorithm. The Scale Saliency algorithm is a visual feature extraction method that extracts local visual salient regions from images. These salient regions are the basis of higher level vision tasks, like object categorization or robot localization (scene categorization). Visual feature extraction algorithms evolved from interest point detection methods, achieving invariance to scale changes and affine transformations.

The first section introduces the concept of interest point and summarizes the evolution of interest point detection algorithms. The algorithms summarized in this section span from the initial and simpler algorithms proposed by Moravec and Förstner to the current state-of-the-art affine and scale invariant visual feature extraction algorithms, like the Maximally Stable Extremal Regions method or the affine invariant version of the Harris corner extraction algorithm.

In the following section we define the concept of local visual saliency, in which the Scale Saliency algorithm is based. The main difference between the Kadir and Brady algorithm and the rest of algorithms presented in this chapter is that it uses Information Theory in order to model local saliency or unpredictability. We will see that this algorithm is a scale-invariant version of

a previous method proposed by Gilles [Gilles, 1998]. The chapter ends with a detailed explanation of the Scale Saliency algorithm.

2.1 Multi-scale and affine visual feature extraction

In this section we introduce the concept of visual feature detection. Starting from the early interest point detection algorithms proposed by Moravec [Moravec, 1977], Förstner [Förstner, 1986] and Harris [Harris and Stephens, 1988], we summarize the evolution of this kind of methods. We will see how these algorithms achieved scale invariance by means of scale-space representations like the one defined by Witkin [Witkin, 1983]. Finally, we will briefly review the current state of the art in visual feature extraction algorithms, in which affine invariance is also achieved.

2.1.1 Interest points

In several Computer Vision problems, instead of processing complete images, the processing is focused on a set of visual features extracted from these images. Visual feature extraction yields a sparse representation of the image. Furthermore, these visual features may in some cases provide additional information. The visual feature extraction topic is present in the Computer Vision literature since thirty years ago. Most of the first visual feature extraction algorithms were aimed to the extraction of edges [Marr and Hildreth, 1980][Canny, 1986][Deriche, 1987]. The work in this thesis is based on a visual feature extraction algorithm that extracts salient regions from images [Kadir and Brady, 2001]. Region extraction algorithms evolved from early works on interest point extraction methods, which performance was high in tasks like stereoscopic matching or object categorization.

An interest point is an image point in which local information is high[1]. These points are highly distinguishable, and thus they are useful in the context of Computer Vision problems. Furthermore, interest points

[1] Another definition of interest point is any local feature for which the signal changes in two dimensions [Schmid and Mohr, 1997].

remain stable under certain image perturbations, like light variations, slight viewpoint changes, and so on. This stability increases the robustness of Computer Vision applications based on interest points. Other advantages may arise from the use of this kind of feature. For instance, in the context of object detection, the extraction of interest points increases robustness to occlusion, due to the fact that interest points are extracted from local parts of the object. Instead of detecting an object as a whole, the object detection algorithm can look for clues of the presence of local parts of the target object in the image.

We can find two types of interest point detectors in the classic literature: geometry-based and intensity-based. In the first case, the detection algorithm usually relies on the previous extraction of geometric features, like edges [Asada and Brady, 1986][Deriche and Faugeras, 1990]. In the second case, interest points are extracted from the image grayscale intensities or from intensity gradients [Moravec, 1977][Rohr, 1990]. We summarize now several of these algorithms. For a more complete review see [Deriche and Giraudon, 1993].

Moravec's corner detector

One of the first interest point detection algorithms was the corner detector proposed by Moravec [Moravec, 1977]. This algorithm computes the difference of grayscale intensities between the pixels in a 3×3 window and the corresponding pixels in other windows shifted 1 pixel towards the eight principal directions.

For each pixel (x, y) on the image, the algorithm builds its 3×3 neighborhood window w and a set $w' = \{w'_{(u,v)}\}$ of shifted windows, being $w'_{(u,v)}$ a 3×3 window centered in $(x + u, y + v)$. The taken values of (u, v) are $\{(1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-1, 1), (0, 1), (1, 1)\}$. The difference between the window w and a shifted window $w'_{(u,v)}$ is given by

$$E_{u,v} = \sum_{i,j \in W} |I_{i,j} - I'_{i,j}|^2 \ , \tag{2.1}$$

where W defines the (i, j) indexes of a 3×3 window, and I and I' are the intensities of the windows w and $w_{(u,v)}$, respectively. The algorithm only

keeps the minimum difference $E = min\{E_{u,v}\}$ for each pixel. The output of the algorithm is the set of local maxima of E that are above a given threshold.

As can be seen, the Moravec's corner detector is a simple and fast algorithm. Its main drawbacks, according to Harris and Stephens [Harris and Stephens, 1988], are three:

- It is an anisotropic detector, due to the fact that the shifts are only considered in $45\,^{\circ}$ intervals.
- The windows are binary and rectangular. As a consequence, the output is noisy. Harris and Stephens suggest to use a smooth circular window (a Gaussian window).
- Only the minimum E is considered for each pixel. Therefore, the algorithm not only detect corners, but also a high amount of edges.

Förstner's corner detector

The corner detector proposed by Förstner [Förstner, 1986] is applied to 5×5 or 7×7 windows. The algorithm computes a covariance matrix for each window, from the grayscale intensity values I of its pixels:

$$C = \hat{\sigma}^2_{\triangle I} \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} = \hat{\sigma}^2_{\triangle I} Q \ . \tag{2.2}$$

In the latter equation, $\hat{\sigma}^2_{\triangle I}$ is the estimation of noise, and I_x and I_y are the derivatives of I along x and y axis, respectively. The covariance matrix represents an ellipse. In order to mark the center of the ellipse (the center of the window) as an interest point, this ellipse should satisfy two constraints: its shape must be close to a circle, and it must be small.

In order to determine how close to a circle the ellipse is, the eigenvalues of the covariance matrix Q are computed. Let $N = Q^{-1}$ be the inverse of Q; the eigenvalues of N and Q are related:

$$\lambda_i(Q) = \frac{1}{\lambda_i(N)} \ . \tag{2.3}$$

Let λ_1 and λ_2 be the eigenvalues of N. In this case, a good estimation of the roundness of the ellipse defined by Q is given by

$$q = \frac{4det(N)}{trace(N)} = \frac{4\lambda_1\lambda_2}{(\lambda_1+\lambda_2)^2} = 1 - \left(\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2}\right)^2 \quad , \tag{2.4}$$

where $det(N)$ is the determinant of N and $trace(N)$ the trace of N. If $q = 0$, then $\lambda_1 = 0$ and $\lambda_2 = 0$, and $det(N) = 0$. Thus, N is singular, meaning that I_x and I_y are linearly dependent: the window is lying over an edge. If $q = 1$, then $\lambda_1 = \lambda_2$. In this case, the covariance matrix Q represents a circle, meaning that the window is over an interest point. It must be noted that in order to estimate q the computation of the eigenvalues of N is not required. The value of q can be estimated from the determinant and the trace of N, that in turn can be computed from $\sum I_x^2$, $\sum I_y^2$ and $\sum I_x I_y$.

Similarly, the trace of Q can be obtained from N:

$$trace(Q) = \frac{trace(N)}{det(N)} \quad . \tag{2.5}$$

The trace of Q is used to estimate the interest value $\hat{w}$ of a window. The interest value is given by

$$\hat{w} = \begin{cases} \frac{1}{trace(Q)} & if \quad q > q_{min} \\ 0 & otherwise \end{cases} \quad , \tag{2.6}$$

where q_{min} is a threshold (usually $q_{min} = 0.5$). The algorithm computes the interest value of the windows centered on all pixels of the image. In the next step (known as non-maximum suppression), the interest value of all non-local maxima is set to zero. All windows for which $\hat{w} \neq 0$ are selected for corner extraction. In each window we select an interest point, computed as the weighted gravitational center of the window. The weight of each pixel is obtained from the direction and the magnitude of its intensity gradient. It must be noted that two or more overlapping windows may yield the same interest point.

The interest value defined in Eq. 2.6 is estimated from local information. This value is not affected by the rest of considered windows. The author proposes a method to identify interest points that are globally significant for the case of images containing textures or repetitive patterns. Let $R = \{r_{ij}\}$ be the correlation matrix obtained from all the detected corners in the image and the grayscale intensities around them. Let

$$S_i = \begin{cases} \frac{1-r_i}{r_i} & if \quad r_i > 0 \\ \infty & otherwise \end{cases} , \tag{2.7}$$

be the seldomness value of a detected corner, and r_i a correlation coefficient that measures the similarity of i with the other corners detected in the image, being

$$r_i = \max_{i \neq j}(r_{ij}) \ . \tag{2.8}$$

In order to use global information, the interest value in Eq. 2.6 is multiplied by S_i (Eq. 2.7).

Harris interest points

One of the best known interest point detectors is the one proposed by Harris and Stephens in 1988 [Harris and Stephens, 1988]. It has been widely cited in the Computer Vision literature, and it has been applied to problems like 3D modeling [Beardsley et al., 1996], object categorization [Leibe et al., 2004] or image mosaicing [Capel and Zisserman, 1998]. This algorithm is based on the the second moment matrix or auto-correlation matrix, which is commonly used to detect features or to describe local structures in images. The matrix is averaged and weighted by means of a Gaussian circular window:

$$A(x, y) = g(\sigma_I) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \ . \tag{2.9}$$

In the latter equation, $g(\sigma_I)$ is a Gaussian kernel with variance σ_I, I_x is the partial derivative of the image I along x direction, and I_y is the partial derivative of the image I along y direction.

The two eigenvalues λ_1 and λ_2 of A represent the two main signal variations in the neighborhood of the pixel (x, y). The kind of feature on which the pixel lies is given by these eigenvalues:

- If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, then the pixel lies on a homogeneous region.
- If $\lambda_1 \approx 0$ and λ_2 is a large positive number, then the pixel lies on an edge.

- If λ_1 and λ_2 are large positive numbers, then the pixel lies on a corner.

Computing the eigenvalues of the second order matrix is computationally expensive. Harris and Stephens propose a metric (usually known as the **Harris function**) in order to quantify the cornerness or corner strength of a point:

$$cornerness = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = det(A) - k \cdot trace(A)^2 \ , \qquad (2.10)$$

where $k \in [0.04, 0.15]$ is a parameter that must be empirically set. The algorithm detects all the pixels for which the *cornerness* value is above a given threshold, after a non-maximum suppression step. An example is given in Fig. 2.1.

Figure 2.1: Example of Harris corners detection. Left: original image. Right: results of the Harris and Stephens algorithm.

Noble's corner detector

In the algorithm proposed by Noble [Noble, 1988] the corners are defined as a conjunction of edges (like in [Cazorla and Escolano, 2003]). The algorithm searches for T, L and X-type edges. This search is based on differential geometry and the topographic surface of the intensity function of the image. Noble states that the Harris interest point detector is only able to detect L-type edges, but no other types of structures. This conclusion is extracted after expressing the Harris function as a function of the First Fundamental Form, that is a representation of the intensity surface of the

image given a small change in x and y. The Noble's approach is based on the Second Fundamental Form. Given the surface defined by the image's intensity function $I(x, y)$

$$S(x, y) = x\mathbf{i} + y\mathbf{j} + I(x, y)\mathbf{k} \ , \tag{2.11}$$

and its unit normal vector

$$\mathbf{n} = \frac{S_x \times S_y}{|S_x \times S_y|} \ , \tag{2.12}$$

that is computed from the derivatives S_x and S_y of S along x and y directions, the equation of the Second Fundamental Form is

$$\Phi_2 = -dS \cdot d\mathbf{n} = \begin{bmatrix} dx & dy \end{bmatrix} \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \ , \tag{2.13}$$

where

$$D = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} S_{xx}\mathbf{n} & S_{xy}\mathbf{n} \\ S_{xy}\mathbf{n} & S_{yy}\mathbf{n} \end{bmatrix} \tag{2.14}$$

and S_{xx}, S_{yy} are the second derivatives of S along the x and y axes, and S_{xy} is the mixed derivative of S in x and y directions. The determinant of D classifies a point according to the local geometry of the intensity surface on that point: for a planar point $L = M = N = 0$, for a parabolic point $LN - M^2 < 0$ and for an hyperbolic point $LN - M^2 > 0$. In the case of absence of noise this classification criterion is sufficient. The interest points are those points classified as hyperbolic or parabolic. But in the case of real-world images this geometric criterion is not enough. Noble proposes a confidence metric based on a statistical analysis of noise. The aim of this confidence metric is to filter false positives, and is given by

$$C = \sqrt{(S_x S_x)(S_y S_y) - (S_x S_y)^2} \frac{|\kappa_1| + |\kappa_2|}{2} \ , \tag{2.15}$$

where κ_1 and κ_2 are two curvature vectors that are orthogonal to the gradient vectors along x and y. All corners which confidence is below a given threshold are discarded.

Wang's corner detector

The algorithm proposed by Wang and Brady [Wang and Brady, 1994] is based on the search of extrema in image surface curvature. Let $I(x, y)$ be the image intensity function. Let also

$$\mathbf{n} = \frac{1}{|\nabla I|}\nabla I \tag{2.16}$$

be the normal vector to the image surface, and $\mathbf{t}$ a unit vector, tangent to the surface and perpendicular to $\mathbf{n}$. The image surface curvature is given by

$$\kappa \approx \frac{\delta^2 I}{\delta \mathbf{t}^2}/|\nabla I| \ , \tag{2.17}$$

where

$$\frac{\delta^2 I}{\delta \mathbf{t}^2} = \frac{1}{\nabla I^2}(I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}) \ . \tag{2.18}$$

In the latter equation I_x and I_y are the first derivatives of I along x and y directions, I_{xx} and I_{yy} are the second derivatives along x and y directions, and I_{xy} is the mixed derivative. As can be seen, curvature is proportional to the second derivative along the edge tangential $\mathbf{t}$ and inversely proportional to edge strength.

The curvature κ is subject to image noise. Image noise can be reduced by means of a Gaussian smoothing kernel $g(\sigma)$. Gaussian smoothing yields a linear displacement of position of the detected corners, proportional to the Gaussian standard deviation. However, for small standard deviation values (for instance, $\sigma = 0.5$), this displacement is smaller than half the size of a pixel and may be ignored. Thus, the algorithm operates on the smoothed image $F = g(\sigma) * I$. We denote the derivatives of F by F_x, F_y, F_{xx}, F_{xy} and F_{yy}.

After introducing these concepts we can summarize the Wang and Brady corner detection algorithm. Firstly, the algorithm performs a non-maximum suppression test based on intensity gradient. The algorithm computes $|\nabla I|$ by means of an edge operator (e.g. the Sobel operator [Sobel and Feldman, 1968]), and selects all those points that are close to large variations in gradient magnitude ($|\nabla I|^2 >> 0$). Selecting this subset of pixels decreases computation time and avoids false positives.

The algorithm searches for pixels for which the curvature exceeds a constant value s. From Eq. 2.17, we can derive the following curvature test:

$$\kappa^2 = \left(\frac{\frac{\delta^2 F}{\delta \mathbf{t}^2}}{|\nabla F|} \right)^2 > s \ . \tag{2.19}$$

After multiplying both sides of this equation by $(|\nabla F|)^2$ we get

$$\left(\frac{\delta^2 F}{\delta \mathbf{t}^2} \right)^2 > s(|\nabla F|)^2 \ , \tag{2.20}$$

and thus:

$$\left(\frac{\delta^2 F}{\delta \mathbf{t}^2} \right)^2 - s(|\nabla F|)^2 > 0 \ . \tag{2.21}$$

In order to be labeled as a corner, a pixel must satisfy the following constraints:

$$\begin{cases} \Gamma = \left(\frac{\delta^2 F}{\delta \mathbf{t}^2} \right)^2 - s(|\nabla F|)^2 = maximum \\ \frac{\delta^2 F}{\delta \mathbf{n}^2} = 0 \\ |\nabla F|^2 > T_1, \Gamma > T_2 \end{cases} , \tag{2.22}$$

where Γ is the local maximum in a $m \times m$ window, s is the curvature constant, and T_1 and T_2 are user defined thresholds.

In the previous equations, the mask of the second derivative is $(-2, -1, 6, -1, -2)$ along the direction of the normal vector to the edge. The mask positions correspond to exact image pixel locations only if the edge is perfectly aligned to x or y axis. In the rest of cases, a subpixel interpolation step must be performed.

Lucas-Tomasi-Kanade feature detector

The Lucas-Tomasi-Kanade feature detector algorithm is an example of interest point detector designed to be applied to a specific problem [Tomasi and Kanade, 1991][Shi and Tomasi, 1994]. The algorithm searches the best features for tracking [Lucas and Kanade, 1981]. Tomasi *et al.* state that the problem of more general approaches is that they are based on a preconceived idea of what an interest point or a corner is, and also that these

approaches are only able to cope with pure translation deformations (affine deformations are ignored). The analysis of the tracking problem led Tomasi *et al.* to a feature extraction algorithm that resembles the Harris algorithm. The main difference is that corner points are selected by means of a greedy test.

The tracking problem involves the image registration of two consecutive video frames $I(x, y, t)$ and $I(x, y, t + \tau)$. The aim of the image registration process is to find a displacement vector $\mathbf{d} = [\xi, \eta]^T$ that satisfies the following equality:

$$J(\mathbf{x}) = I(\mathbf{x} - \mathbf{d}) + n(\mathbf{x}) \ , \tag{2.23}$$

where $J(\mathbf{x}) = I(x, y, t + \tau)$ and $I(\mathbf{x} - \mathbf{d}) = I(x - \xi, y - \eta, t)$ and $n(\mathbf{x})$ is a noise term. Usually the difference between $J(\mathbf{x})$ and $I(\mathbf{x} - \mathbf{d})$ is not only due to displacement, but also to other kinds of transformations (lighting condition variations, occlusions, affine distortions, and so on). Thus, the image registration process searches the displacement vector $\mathbf{d}$ that minimizes the error

$$\epsilon = \int_W [I(\mathbf{x} - \mathbf{d}) - J(\mathbf{x})]^2 w(\mathbf{x}) \, d\mathbf{x} \ , \tag{2.24}$$

where W is an image window or region of interest and $w(\mathbf{x})$ is a weight function relative to W (e.g. a Gaussian kernel). If the displacement $\mathbf{d}$ between frames is small enough, the intensity function can be approximated by its Taylor series truncated to the first term

$$I(\mathbf{x} - \mathbf{d}) = I(\mathbf{x}) - \mathbf{g} \cdot \mathbf{d} \ , \tag{2.25}$$

where $\mathbf{g} = [I_x, I_y]^T$ is a vector of intensity gradients. If this approximation is plugged into Eq. 2.24 we get

$$\epsilon = \int_W [h - \mathbf{g} \cdot \mathbf{d}]^2 w(\mathbf{x}) \, d\mathbf{x} \ , \tag{2.26}$$

where $h = I(\mathbf{x}) - J(\mathbf{x})$. In order to minimize the residual ϵ, we differentiate Eq. 2.26 with respect to $\mathbf{d}$ and we set the result equal to zero

$$\int_W (h - \mathbf{g} \cdot \mathbf{d}) \mathbf{g} w(\mathbf{x}) \, dA = 0 \ . \tag{2.27}$$

From the latter equation, knowing that $(\mathbf{g} \cdot \mathbf{d})\mathbf{g} = (\mathbf{g}\mathbf{g}^T)\mathbf{d}$ and assuming that $\mathbf{d}$ is constant in W, we derive

$$(\int_W \mathbf{g}\mathbf{g}^T w(\mathbf{x}) dA)\mathbf{d} = \int_W h\mathbf{g} w(\mathbf{x}) dA \ , \tag{2.28}$$

that is a system of two equations with two unknowns. The system can be rewritten as

$$G\mathbf{d} = \mathbf{e} \ . \tag{2.29}$$

The G matrix has the form of the second moment matrix (Eq. 2.9) used in the Harris algorithm [Schmid et al., 2000] and $\mathbf{e}$ is an error vector. Tomasi *et al.* state that given a window W centered in a point $\mathbf{x}$, this point is an interest point if its corresponding matrix G is above noise level and well-conditioned. The first constraint implies that both eigenvalues λ_1 and λ_2 of G are large. The second constraint implies that both eigenvalues should not differ by several orders of magnitude. In practice, an interest point must satisfy

$$\min(\lambda_1, \lambda_2) > \lambda \ , \tag{2.30}$$

being λ a given threshold. Due to the fact that the range of intensity values in an image is not wide, if the lowest eigenvalue is large then the highest eigenvalue will also be large and their difference will be small. The lower bound of λ is estimated from the eigenvalues of a homogeneous region. The upper bound of λ is extracted from the eigenvalues of several salient features like corners or textured regions. The threshold λ is set in the range between these bounds; however, this parameter is not critical.

SUSAN

The SUSAN algorithm [Smith and Brady, 1995] is a simple and fast algorithm based on a non-linear filter. Contrary to previous described approaches, this algorithm does not compute neither Gaussian derivatives nor image gradients. A circular mask of radius 3.4 (an area of 37 pixels) is centered in each point of the image. The center of the mask is the nucleus, and the area inside the mask with a similar intensity to that of the nucleus is called USAN (Univalue Segment Assimilating Nucleus) area. We may classify the

nucleus according to the USAN area. If it occupies half of the mask area the nucleus is an edge point, and if it occupies less than half of the mask area it is a corner point. The output of the algorithm is a set of SUSAN (Smallest Univalue Segment Assimilating Nucleus) corner points.

The first step involves the computation of the USAN area for each nucleus, that is, the number of pixels inside the mask which intensity is similar to that of the nucleus:

$$n(\mathbf{x}_0) = \sum_{\mathbf{x}} c(\mathbf{x}, \mathbf{x}_0) \ , \tag{2.31}$$

where $\mathbf{x}_0$ is the nucleus position and $\mathbf{x}$ is the position of the other points in the mask and c is a function that determines if a point is part of the USAN area. A simple version of this function is given by

$$c(\mathbf{x}, \mathbf{x}_0) = \begin{cases} 1 & if \quad |I(\mathbf{x}) - I(\mathbf{x}_0)| \leq t \\ 0 & otherwise \end{cases} \ , \tag{2.32}$$

where t is a threshold and I the intensity function of the image. The intensity of a point is similar to that of the nucleus if the difference between both is below t. The authors also proposed a more stable version of c:

$$c(\mathbf{x}, \mathbf{x}_0) = e^{-\left(\frac{I(\mathbf{x}) - I(\mathbf{x}_0)}{t}\right)^6} . \tag{2.33}$$

After the USAN area is calculated, the *cornerness* of all pixels is estimated by means of the application of a geometric constraint: only those pixels for which $n(\mathbf{x}_0) < g$, being $g = n_{max}/2$ and n_{max} the maximum possible value of n are corner candidates. The cornerness is computed as the difference between the USAN area and the geometric threshold g:

$$R(\mathbf{x}_0) = \begin{cases} g - n(\mathbf{x}_0) & if \quad n(\mathbf{x}_0) < g \\ 0 & otherwise \end{cases} \ . \tag{2.34}$$

Two tests are applied in order to avoid false positives. The first one is based on the gravity center of the USAN area. The corner candidates for which this center is close to the nucleus are rejected. The second one is based on the intensity of the pixels that lie on the line between mask boundaries that passes through the nucleus and the gravity center. If any of these pixels

is not part of the USAN area, the corner candidate is also rejected. These tests are aimed to decrease the effect of blurred boundaries and noise.

Finally, the algorithm performs a non-maximum suppression step.

Minimum Intensity Change

The Minimum Intensity Change algorithm proposed by Trajkovic *et al.* [Trajkovic and Hedley, 1998] is a fast corner detection algorithm. It is based on the variation of intensity along arbitrary lines that pass through the studied point. The algorithm uses the terminology of the SUSAN method in order to define a cornerness function. Let l be a line that contains the nucleus; it intersects with two opposite points $\mathbf{x}$ and $\mathbf{x}'$ in the boundaries of the discretized circular window W. The cornerness function is then

$$R(\mathbf{x}_0) = \min_{\mathbf{x},\mathbf{x}' \in W} ((I(\mathbf{x}) - I(\mathbf{x}_0))^2 + (I(\mathbf{x}') - I(\mathbf{x}_0))^2) \ , \tag{2.35}$$

where $I(\mathbf{x}_0)$ is the intensity of the window nucleus, and $I(\mathbf{x})$ and $I(\mathbf{x}')$ are the image intensities of points $\mathbf{x}$ and $\mathbf{x}'$, respectively. Given this definition, there are three possible cases:

- the nucleus is in a homogeneous region. Then, there is at least a line l for which both $\mathbf{x}$ and $\mathbf{x}'$ are part of the USAN area. Thus, the value of $R(\mathbf{x}_0)$ is low.
- the nucleus is in an edge. There is exactly one line l (tangential to the edge) for which both $\mathbf{x}$ and $\mathbf{x}'$ belong to the USAN area. Once again, the value of $R(\mathbf{x}_0)$ is low.
- the nucleus is a corner point. For any line l, at least one of the points $\mathbf{x}$ or $\mathbf{x}'$ do not belong to the USAN area. Therefore, the value of $R(\mathbf{x}_0)$ is high.

The main issue of this method is that it can label any pixel in a strong edge can be labeled as a corner if this edge is not tangent to any of the studied line directions. Instead of increasing the size of the circular window (a large window could deteriorate the quality of the corner detection), the algorithm performs a subpixel interpolation.

Prior to corner detection, the algorithm applies a test based on the horizontal and vertical intensity variations (R_H and R_V, respectively):

$$R_H = (I(\mathbf{x}_H) - I(\mathbf{x}_0))^2 + (I(\mathbf{x}'_H) - I(\mathbf{x}_0))^2 \ , \tag{2.36}$$

$$R_V = (I(\mathbf{x}_V) - I(\mathbf{x}_0))^2 + (I(\mathbf{x}'_V) - I(\mathbf{x}_0))^2 \ . \tag{2.37}$$

In the latter equations, $I(\mathbf{x}_H)$ and $I(\mathbf{x}'_H)$ are the opposite points in the boundaries of W for the horizontal line l that passes through the nucleus, and $I(\mathbf{x}_V)$ and $I(\mathbf{x}'_V)$ are the opposite points in the boundaries of W for the vertical line l that passes through the nucleus. The test is defined as:

$$R(\mathbf{x}_0) = min(R_H, R_V) \ . \tag{2.38}$$

If $R(\mathbf{x}_0)$ is below a given threshold, then $\mathbf{x}_0$ is not a corner and it is discarded. Otherwise, subpixel interpolation is applied in order to search diagonal edges.

The algorithm also filters false positives by means of multi-resolution analysis. The aim of this step is to reject texture corners. The authors consider two types of corners: geometric and texture corners. Geometric corners correspond to real corners of objects in the image. These corners are correctly detected at different scales, but their number is low. By the other hand, texture corners are those detected on small or textured objects. They do not correspond to real corners. Furthermore, their density is higher. Thus, texture corners are not suitable for matching.

The steps of the Minimum Intensity Change algorithm are:

- Apply Eq. 2.38 to the low resolution version of the image. All those points for which $R(\mathbf{x}_0)$ is higher than a given threshold T_1 are labeled as potential corners. The rest are discarded.

- Apply Eq. 2.38 to the potential corners in the full resolution image. If $R(\mathbf{x}_0)$ is higher than a threshold T_2 for a given potential corner its cornerness is computed using Eq. 2.35 and subpixel interpolation.

- Non-maximum suppression.

Experimental results demonstrate that the algorithm's accuracy is similar to those of the Harris, Wang or SUSAN algorithms. The advantage of Minimum Intensity Change is its speed. The algorithm is remarkably faster than the aforementioned methods.

FAST algorithm

The FAST (Features from Accelerated Segment Test) algorithm proposed by Rosten and Drummond [Rosten and Drummond, 2006] is a recent corner detection approach based on Machine Learning and Information Theory. The motivation of their work was to build a corner detection algorithm able to operate in real time. In order to do that, the FAST algorithm is based on a fast filter called *segment test*. The FAST algorithm not only is faster than other approaches like Harris, SUSAN, and so on, but also it detects higher-quality corners.

The segment test considers a circle formed by 16 pixels around the corner candidate $\mathbf{p}$. The pixel $\mathbf{p}$ is labeled as a corner if a set of n contiguous pixels in the circle are brighter than $\mathbf{p}$ (their intensity is higher than $I(\mathbf{p}) + t$, being t a threshold) or are darker than $\mathbf{p}$ (their intensity is lower than $I(\mathbf{p}) - t$). If the value $n = 12$ is chosen, the test is even faster: only the pixels 1, 5, 9 and 13 shown in Fig. 2.2 must be tested. If at least three of these pixels are brighter or darker than $\mathbf{p}$, this point $\mathbf{p}$ is labeled as a corner. Otherwise it is discarded.

This approach is limited, due to its poor generalization for $n \neq 12$ and the fact that using pixels 1, 5, 9 and 12 assumes a certain corner structure. Rosten and Drummond proposed a Machine Learning algorithm in two phases to overcome these issues. The first phase extracts information from a set of training images, and the second phase builds a classifier using Information Theory.

Phase 1. After selecting the values of t and n, the complete segment test is applied to a set of training images (the n circle pixels are tested for each pixel in these training images). Each pixel $x \in \{1, \ldots, n\}$ in the circle, in a position $P \to x$ relative to $\mathbf{p}$, is labeled depending on its intensity $I(P \to x)$:

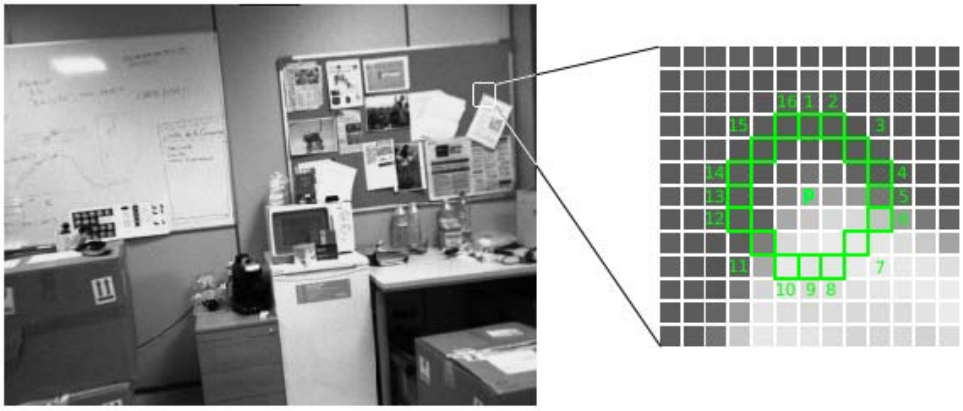


Figure 2.2: Example of application of the segment test to a pixel p in the image shown in Fig. 2.1.

$$S(P \to x) = \begin{cases} d, & & I(P \to x) & \leq & I(\mathbf{p}) - t & (darker) \\ s, & I(\mathbf{p}) - t & < & I(P \to x) & < & I(\mathbf{p}) + t & (similar) \\ b, & I(\mathbf{p}) + t & \leq & I(P \to x) & & & (brighter) \end{cases} . \tag{2.39}$$

The computation of $S(P \to x)$ for a given pixel x and all $\mathbf{p} \in P$ (the set of all pixels in the training images) splits P into three different sets P_d, P_s and P_b. These sets will be used during phase 2 in order to build the corner classifier.

Phase 2. Let k_p be a boolean variable which value is *true* if $\mathbf{p}$ is a corner and $false$ otherwise. The algorithm selects the circle pixel x that provides more information about the value of k_p for any candidate pixel $\mathbf{p}$. The entropy of k_p for the set P is

$$H(P) = (c + \bar{c}) \log_2(c + \bar{c}) - c \log_2 c - \bar{c} log_2 \bar{c} \ , \tag{2.40}$$

being

$$c = |\{\mathbf{p} | k_p = true\}| \tag{2.41}$$

the number of corners and

$$\bar{c} = |\{\mathbf{p}|k_p = false\}| \tag{2.42}$$

the number of non-corners. The algorithm selects the pixel x that maximizes the information gain given by

$$H(P) - H(P_d) - H(P_s) - H(P_b) \ . \tag{2.43}$$

This process is recursively applied to the three subsets of P, searching a x_b that splits P_b into three subsets $P_{b,d}$, $P_{b,s}$ and $P_{b,b}$, a x_s that splits P_s into $P_{s,d}$, $P_{s,s}$ and $P_{s,b}$ and a x_d that splits P_d into $P_{d,d}$, $P_{d,s}$ and $P_{d,b}$. The termination condition is satisfied when the entropy of a subset is zero, meaning that all the points $\mathbf{p}$ in that subset share the same value of k_p (all the points in the subset are corners or non-corners). The result of this recursive procedure is a classifier tree that labels a point $\mathbf{p}$ as a corner or a non-corner. The tree is optimized by means of branch reordenation and redundancy removal.

Non-maximum suppression can not be directly applied, due to the fact that the algorithm is not based on calculating a cornerness value for each point. In order to apply non-maximum suppression, a score V is assigned to each detected corner:

$$V = \max\left(\sum_{x\in S_{bright}} |I(P\to x) - I(\mathbf{p})| - t, \sum_{x\in S_{dark}} |I(\mathbf{p}) - I(P\to x)| - t\right) \ , \tag{2.44}$$

where

$$S_{bright} = \{x|I(P\to x) \geq I(\mathbf{p}) + t\} \ , \tag{2.45}$$

$$S_{dark} = \{x|I(P\to x) \leq I(\mathbf{p}) - t\} \ . \tag{2.46}$$

Using the trained classifier for $n = 9$, only an average of two or three questions are required to classify a pixel. The main drawbacks of the algorithm are that it is sensitive to noise (a small amount of pixels per candidate are tested) and that it detects corners in one-pixel-wide edges.

2.1.2 The scale-space representation

The interest point detectors presented in the previous section are based on operators that are applied to a fixed size region of the image. the consequence is that interest points are only detected in a narrow range of scales. However, the characteristic features in real-world images may be found at different scales. Furthermore, the scale of these image features can provide additional information to subsequent vision processes. If we focus on an only scale we are losing important information about the image. The scale-space representation emerged as an approach to detect the presence of local structures at different scales.

A precursor of this representation is the Gaussian pyramid of an image. It is an early method to process images at multi-scale level. The filter designed by Burt [Burt, 1981] successively applies a Gaussian smoothing operator and a sub-sampling step to an image in order to generate a set of smoothed images at different resolutions. This method can be recursively defined as:

$$\begin{cases} G_0(x,y) = I(x,y) \\ G_l(x,y) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) G_{l-1}(2x+m, 2y+n) \end{cases}, \tag{2.47}$$

where l is the image level in the pyramid, $I(x,y)$ the original intensity function of the image, and $w(x,y)$ a weighting function, usually a Gaussian kernel, equally applied to all levels. The image resolution decreases for each level. Lower resolution images will be focused on coarser features. The name of this method is due to the fact that a graphical representation of the complete process resembles a pyramid of decreasing resolution images.

The concept of **scale-space** was first introduced by Witkin [Witkin, 1983], who proposed a method to build a qualitative description of a 1D signal at various scales. Given a 1D signal, and independently of the interpretation given to its extrema (they may indicate, for instance, the presence of edges in an image), its scale can be modified by Gaussian smoothing. However, scale changes introduce ambiguity, due to the fact that in each scale previous extrema disappear and new extrema are created. Witkin states that these different scales should not be individually considered. The information at different scales must be somehow related. This is exactly what his scale-space

representation achieves: it yields a tree that qualitatively represents the signal at any scale of observation.

In order to build the representation of a 1D signal $f(x)$ at scale σ, this signal is convolved with a Gaussian kernel $g(\sigma)$:

$$L(x, \sigma) = g(\sigma) * f(x) \ . \tag{2.48}$$

The result of smoothing the signal with different σ values is a surface that represents the scale-space of the signal. The Gaussian kernel is applied because, among other interesting properties, it is the only one for which as σ decreases new zero crossings may appear, but not disappear [Babaud et al., 1986]. This feature of the Gaussian kernel is important, due to the fact that the scale-space representation focuses on zero crossings of the second derivative $L_{xx} = 0$ for which the third derivative satisfies $L_{xxx} = 0$.

The contours of $L_{xx} = 0$ in the scale-space show the presence of signal events or local structures of interest. These contours are called zero contours. All the extrema of $f(x)$ at different scales belonging to the same zero contour are considered part of the same underlying event. The actual location of this event can be obtained using the scale $\sigma = 0$. Thus, coarser scales are used to detect signal extrema, and the finest scale is used to correctly locate these extrema. Any detected event is defined as a pair (x, σ), being x its location and σ its scale.

Although event tracking from coarser to finer scales solves the problem of finding these events, their description still depends on the chosen scale. Witkin proposed a tree representation of the scale-space that simultaneously represents the signal and its events at different scales. He also defined a stability criterion, aimed to filter non-interesting events. This criterion removes those events that are not present in a wide range of scales.

In his work Witkin suggested that the scale-space representation could be extended to the domain of 2D signals, like images. In this case, the scale-space would be a volume containing zero-crossing surfaces. Yuille and Poggio [Yuille and Poggio, 1986] were the ones who finally extended Witkin's scale-space representation to the domain of 2D data, and in general to any data dimensionality. They proved that for 2D data the 2D rotationally

symmetric Gaussian mask is the only scaling filter that is well behaved, that is, the only scaling filter for which any zero crossings is destroyed in the scale-space. This feature of the Gaussian kernel allows the coarse to fine scale tracking of the interest features in an image.

The same conclusions were reached by Koenderink and Richards [Koenderink and Richards, 1988]. They state that building the scale-space representation by means of Gaussian smoothing is equivalent to find the solution to the diffusion equation

$$\frac{\partial L}{\partial t} = \frac{1}{2}\nabla^2 L = \frac{1}{2}\sum_{i=1}^{D} \partial x_i x_i L \ , \tag{2.49}$$

where t is the scale and D is the number of dimensions. In the case of 2D data:

$$\frac{\partial L}{\partial t} = \frac{1}{2}(L_{xx} + L_{yy}) \ . \tag{2.50}$$

In the diffusion equation (Eq. 2.49) the intensity distribution of the image represents a temperature distribution. The scale-space of order t represents the heat diffusion along the plane of the image over time t, assuming a constant value of $1/2$ for the material conductivity.

The main issue of all these works is that they not provide a method to select the appropriate local scale of the image features. In this sense, Lindeberg proposed a *general* framework for the selection of the characteristic scales of the local features in an image. This framework is based on the detection of local maxima over scales and it can be applied to different feature types, like blobs, corners, edges or junctions [Lindeberg, 1998][Lindeberg, 1994].

Let $D_{norm}L$ be a normalized differential filter that responds to a given type of feature in the scale-space. Lindeberg states that the simultaneous detection of the scale and the location of an image feature can be achieved by means of a search of local extrema in both image and scale-space:

$$\begin{cases} (\nabla(D_{norm}L))(x,y) = 0 \\ (\frac{\partial D_{norm}L}{\partial t})(x,y) = 0 \end{cases} \ . \tag{2.51}$$

This method makes possible the detection of **scale-invariant** features. If the image is scaled using a constant scale factor σ, then the characteristic scale

of its local features will be also multiplied by the same factor σ. Lindeberg states that the location of a local feature in a coarse scale may be incorrect and that a two-phase detection approach must be applied. Firstly, the features are detected in coarser scales, and then their location is refined using information from finer scales.

In this section we summarize several visual feature extraction algorithms that rely on multi-scale information. The two first approaches yield interest points that are not scale-invariant. The rest of algorithms detect scale-invariant visual features. For more information about scale-space see [Lindeberg, 2008].

Deriche and Giraudon

In the interest point detector proposed by Deriche and Giraudon [Deriche and Giraudon, 1993], the scale-space paradigm is applied to improve the location accuracy of the detected corners (but not to detect scale-invariant interest points). They state that previous algorithms, like the Harris interest point detector [Harris and Stephens, 1988], are not able to locate the exact position of corners, even in the case of synthetic images.

Deriche and Giraudon studied a corner model and its behavior in the scale-space and they reached two conclusions: the exact position of a corner can be detected as an stable zero crossing in scale-space, and a local maximum of the measures defined by Beaudet and Nagel (see below) moves along the bisector line that passes through the exact position of the corner point.

The cornerness measure by Beaudet [Beaudet, 1978] is a rotation invariant operator defined as follows:

$$DET = I_{xx}I_{yy} - I_{xy}^2 \ , \tag{2.52}$$

being DET the determinant of the Hessian matrix

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \ , \tag{2.53}$$

where I_{xx} and I_{xy} are the second order derivatives of the image I along axis x and y respectively, and I_{xy} is the mixed derivative in x and y directions.

The corner detection algorithm by Beaudet operates in two steps. Firstly, all pixels for which DET is lower than a given threshold are filtered. Then the algorithm performs a non-maximum suppression step.

The method proposed by Nagel [Nagel, 1983] is based on the Gaussian curvature, a concept first defined by Lipschutz [Lipschutz, 1969] that relates H with the product of the principal curvatures $\kappa_{min}\kappa_{max}$

$$(\kappa_{min}\kappa_{max}) = \frac{DET}{(1 + I_x^2 + I_y^2)^2} \quad , \tag{2.54}$$

being I_x and I_y the first derivatives of I along x and y directions. The product of the principal curvatures can be used to classify a pixel in terms of differential geometry:

- if $\kappa_{min}\kappa_{max} > 0$, the pixel is an elliptic point.
- if $\kappa_{min}\kappa_{max} < 0$, the pixel is a hyperbolic point.
- if $\kappa_{min}\kappa_{max} = 0$, the pixel is a parabolic point.

The Nagel's algorithm [Nagel, 1983] works as follows: firstly it applies Eq. 2.54 to all pixels on the image, and then it matches the position of each elliptic point e with the position of a hyperbolic point h. The direction of the curvatures in e and h must have opposite signs and they must also be approximately aligned. The last step consists in selecting, for each pair of matched points, the point t for which its main curvature is equal to zero. This is the maximum slope point on the surface defined by the image between e and h.

Now we summarize the Deriche and Giraudon algorithm, that is based on the previous concepts:

- Firstly, two images DET_1 and DET_2 are built from the response of the DET measure (Eq. 2.52) for two different scales $\sigma_1 < \sigma_2$ in the scale-space.
- The positive local maxima of DET_1 and DET_2 are detected.
- Each local maxima (x_1, y_1) in DET_1 is matched to the nearest local maxima (x_2, y_2) in DET_2, following a spiral path defined by a 7×7 window.

- In order to improve corner location, the algorithm fits a quadratic surface around each local maxima. Then the exact position of the corner, at subpixel level, can be selected. This step is stable except for corners which angle is lower than $\pi/4$. In this case larger neighborhoods should be used.

- The final step computes the equation of the line between (x_1, y_1) and (x_2, y_2). This line is an estimation of the bisector line, on which the exact location of the corner can be found. The corner is located in the first zero crossing of the Laplacian along this line, from (x_2, y_2) to (x_1, y_1), at subpixel level.

Wavelet-based points

The wavelet-based points algorithm proposed by Loupias *et al.* [Loupias et al., 2000][Sebe and Lew, 2001] detects non-scale-invariant interest points by means of a multi-scale analysis based on wavelet transforms [Mallat, 1989][Stollnitz et al., 1995]. The algorithm was applied to the problem of natural image retrieval. Previous interest point detectors presented two main drawbacks when applied to natural images: the salient features in natural images are not necessarily corner-like and most interest points that are detected in textured regions of the image are not globally salient. The consequence of these drawbacks is that the detected points are not well distributed across the image, degrading the quality of the image representation and the image retrieval process. The wavelet-based points algorithm detects globally salient points in the image that may or not correspond to corners.

Wavelet transforms detect image variations at coarse and fine resolutions. A wavelet is an oscillating and attenuated function which integral is equal to zero. An image I is studied at a set of scales $\{2^j\}$, being $j \leq -1$. The wavelet detail image $W_{2^j}I$ is the result of convolving the image I with the corresponding wavelet at scale 2^j. This wavelets are orthogonal and lead to a complete and non-redundant representation of the image.

The aim of the algorithm is to search parts of the image image that are salient at any scale. A high absolute value of the wavelet coefficient $W_{2^j}I(\mathbf{x})$ in a coarse scale represents an image region with high global variation. The

location of the salient point is refined using wavelet coefficients at lower scales. Due to use of wavelets with compact support (an example is the Haar wavelet function, that is the simplest and fastest one), the point from which a wavelet coefficient at scale 2^j was computed is known, and also it is possible to study its coefficients at a finer scale 2^{j+1}.

The set of coefficients at scale 2^{j+1} that are computed from the same point that the coefficient $W_{2^j}I(\mathbf{x})$ at scale 2^j are called children $C(W_{2^j}I(\mathbf{x}))$ of $W_{2^j}I(\mathbf{x})$. Each wavelet coefficient $W_{2^j}I(\mathbf{x})$ is calculated from a set of $2^{-j}p$ points of the image, and it represents image variation at scale 2^j. Its children coefficients represent the variation of subsets of these points. The most salient subset is the one with highest wavelet coefficient at scale 2^{j+1}, that is, the maximum absolute value in the set $C(W_{2^j}I(\mathbf{x}))$.

The algorithm searches for the highest child coefficient of each wavelet coefficient at the coarsest scale, and then it recursively repeats this step until a coefficient $W_{2^{-1}}I(\mathbf{x})$ at the finest scale is selected. This coefficient represents a small subset of points, from which the point with highest gradient is selected as salient point. The saliency value of this salient point is the sum of the absolute value of wavelet coefficients tracked, from the coarsest to the finest scale, in order to detect it:

$$saliency = \sum_{k=1}^{-j} \left| C^{(k)}\left(W_{2^j}I(\mathbf{x})\right)\right|, 0 \leq n < 2^j N, -J_{max} \leq j \leq -1 \ . \quad (2.55)$$

Finally, all these salient points for which their saliency is below a given threshold are rejected.

Laplacian of Gaussian

The visual feature extraction algorithm proposed by Blostein *et al.* [Blostein and Ahuja, 1989] was the first multi-scale blob detector based on the search of maxima of the Laplacian function in the scale-space. Given the image intensity function $I(x, y)$, the response to the $\nabla^2 G$ function is

$$\nabla^2 G(x,y) * I(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{2\sigma^2 - (u^2+v^2)}{\sigma^4} e^{\frac{-(u^2+v^2)}{2\sigma^2}} I(x-u, y-v)dudv \ . \quad (2.56)$$

The image of a disk of diameter D and intensity C is given by:

$$I(x,y) = \begin{cases} C\ if\ x^2+y^2 \leq \frac{D^2}{4} \\ 0\ otherwise \end{cases} \quad . \tag{2.57}$$

Plugging the latter equation in Eq. 2.56 we obtain the response of $\nabla^2 G$ on the center of the disk when $x = 0$ and $y = 0$:

$$\nabla^2 G\ response = \frac{\pi C D^2}{2\sigma^2} e^{\frac{-D^2}{8\sigma^2}} \quad , \tag{2.58}$$

$$\frac{\partial}{\partial\sigma}\nabla^2 G\ response = \frac{\pi C D^2}{2}\left(\frac{D^2}{4\sigma^5} - \frac{2}{\sigma^3}\right) e^{\frac{-D^2}{8\sigma^2}} \quad . \tag{2.59}$$

If we work out the diameter D and the intensity C of the disk from the latter equations we get

$$D = 2\sigma\sqrt{\frac{\sigma\left(\frac{\partial}{\partial\sigma}\nabla^2 G * I\right)}{(D^2 G * I) + 2}} \quad , \tag{2.60}$$

$$C = \frac{2\sigma^2}{\pi D^2} e^{\frac{-D^2}{8\sigma^2}} (\nabla^2 G * I) \quad . \tag{2.61}$$

Now we can summarize the steps of the algorithm. First, it applies the convolutions $\nabla^2 G * I$ and $(\partial/\partial\sigma)\nabla^2 G * I$ to the image I, using the set of scales $\sigma = \{i\sqrt{2} | i = 1 \ldots 6\}$. In order to compute $\nabla^2 G * I$ for a given σ, the image is convolved using a discretized mask computed as

$$\frac{2\sigma^2 - r^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}} \quad , \tag{2.62}$$

where r is the radius of the mask. And in order to compute $(\partial/\partial\sigma)\nabla^2 G * I$ for a given σ value, the image is convolved with a discretized mask computed as

$$\frac{6r^2\sigma^2 - r^4 - 4\sigma^4}{\sigma^7} e^{-\frac{r^2}{2\sigma^2}} \quad . \tag{2.63}$$

Next, the algorithm searches the local maxima and minima of $\nabla^2 G * I$ using a 3×3 window. The algorithm will try to fit disks in these locations. A local maxima corresponds to a possible disk which intensity is higher than that of its surrounding pixels, and a local minima corresponds to a

disk which intensity is lower than that of its surrounding pixels. For each candidate location, the diameter and the intensity of the disk is estimated using Eq. 2.60 and Eq. 2.61. Finally, a candidate is labeled as visual feature only if $w-2 \leq D \leq w+2$, where w is the diameter of the filter $\nabla^2 G$ for which the candidate was selected.

Harris-Laplace

The Harris-Laplace algorithm proposed by Mikolajczyk and Schmid [Mikolajczyk and Schmid, 2001] is an example of interest point detector adapted to the scale-space paradigm in order to achieve scale invariance. The scale-space representation is built following the method proposed by Witkin [Witkin, 1983], being the result a set of images at different resolutions. These different resolution levels are computed by the convolution of the image with a Gaussian kernel with increasing variance σ:

$$L(x, y, \sigma) = g(\sigma) * I(x, y) \ . \tag{2.64}$$

The multi-scale adapted second moment matrix (see Eq. 2.9) is

$$\mu(x, y, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} L_x^2(x, y, \sigma_D) & L_x L_y(x, y, \sigma_D) \\ L_x L_y(x, y, \sigma_D) & L_y^2(x, y, \sigma_D) \end{bmatrix} \ , \tag{2.65}$$

where L_x is the first derivative of L along x direction, L_y is the first derivative of L along y direction, σ_I is an integration scale, and σ_D is a differentiation scale. The multi-scale adapted Harris function is

$$cornerness = det(\mu(x, y, \sigma_I, \sigma_D)) - k \cdot trace(\mu(x, y, \sigma_D, \sigma_I))^2 \ . \tag{2.66}$$

The algorithm applies Eq. 2.66 to all image pixels in a given range of scales. The scale factor is exponentially distributed. The scale in the resolution level n is calculated as

$$\sigma_{I_n} = l^n \sigma_{I_0} \ , \tag{2.67}$$

where σ_{I_0} is the initial scale, σ_{I_n} the scale at level n, and l the scale variation factor between successive resolution levels. Regarding σ_D, it is derived from σ_I following the equation $\sigma_D = s\sigma_I$, where s is a constant value.

In order to estimate the characteristic scale of the visual features, the algorithm searches for an extrema of a given function over scales [Lindeberg, 1998]. The experimental results obtained by Mikolajczyk and Schmidt demonstrated that the best results are obtained with Laplacian over Gaussian (LoG) function

$$det(LoG(x, y, \sigma_{I_n})) = \sigma_{I_n}^2 det(L_{xx}(x, y, \sigma_{I_n}) + L_{yy}(x, y, \sigma_{I_n})) \ , \qquad (2.68)$$

where L_{xx} is the second derivative of L along x direction and L_{yy} is the second derivative of L along y direction.

The main drawback of the Harris-Laplacian method is that, for a given visual feature, the functions in Eq. 2.66 and Eq. 2.68 attain several maxima at different scales **and** locations. An iterative algorithm later proposed by Mikolajczyk and Schmid [Mikolajczyk and Schmid, 2004b] tunes location and scale simultaneously from an initial set of multi-scale interest points. As a consequence, several of these maxima converge. An example of application is shown in Fig. 2.3.

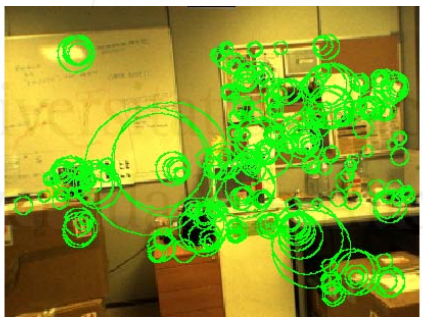


Figure 2.3: Example of application of the Harris-Laplace algorithm to the image in Fig. 2.1.

Difference of Gaussians

The SIFT (Scale Invariant Feature Transform) algorithm is a feature extraction algorithm that assigns a descriptor to each extracted feature [Lowe, 1999][Lowe, 2004]. The algorithm is divided into two phases. In the first one, the algorithm searches interest points that are

stable in the scale-space. In the second phase, a local descriptor vector is computed for each of these interest points. The SIFT descriptor (the second part of the algorithm) has been widely applied in the literature, due to its good performance when compared with other descriptors [Mikolajczyk and Schmid, 2004a]. In this survey we will focus on the first phase of the algorithm, commonly known as DoG (Difference of Gaussians). See [Lowe, 2004] for a complete analysis of the SIFT descriptor.

The first step involves searching for extrema of the DoG function in the scale-space $L(x, y, \sigma)$, that is built based on Eq. 2.64. These extrema define the location and scale of candidate points (several of these candidate points will be filtered during later stages of the algorithm). The $DoG(x, y, \sigma)$ function is computed as the difference between two images in the scale-space, separated by a multiplication factor of k:

$$DoG(x, y, \sigma) = (g(k\sigma) - g(\sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \ . \quad (2.69)$$

The DoG is used in the algorithm due to the fact that it is computationally efficient and that it approximates the Laplacian of Gaussian function, normalized with respect to scale. Lowe stated that the normalization of the Laplacian with respect to σ^2 is necessary in order to achieve scale invariance [Lowe, 2004].

The scale-space is represented by means of a pyramid of images at different resolutions. This pyramid is split into octaves. Inside each octave, the image of the previous level of the pyramid is convolved with a Gaussian in order to obtain a new image separated by a factor of k in the scale-space. Each octave is composed of s levels, thus $k = 2^{(1/s)}$. The first image in an octave is computed from the last image in the previous octave, after resampling it by means of a bilinear interpolation. The DoG function is applied to adjacent smoothed images in the pyramid. A pixel is labeled as candidate point if its DoG value is higher or lower than the value of its 8 neighbors in the same level and than that of its 9 neighbors both in the previous and next level in the scale-space pyramid. After this step, the amount of detected candidate points is high. Therefore, Lowe proposed two additional criteria in order to select the most stable and useful of these candidate points.

Firstly, the algorithm estimates the actual interpolated location of the detected extrema in $DoG(x, y, \sigma)$. A 3D quadratic function is fitted to the local set of sample points, by means of a Taylor expansion of the scale-space function, shifted so the origin is on the candidate point. Then, if the DoG value in this interpolated position is lower than a given threshold, the candidate point is discarded. This step filters low-contrast candidate points that are sensitive to noise. The algorithm also filters all the candidate points that lie on an edge. This step, that resembles the Harris algorithm, estimates the principal curvatures of the candidate point from the eigenvalues of the Hessian Matrix (see Eq. 2.53). If the ratio of the eigenvalues is below a given threshold, the candidate point is discarded.

The main drawback of the interest point detection step of the SIFT method is that it is not affine-invariant (see Section 2.1.3). Lowe states that affine invariance is achieved during the descriptor computation step, instead of during feature extraction.

Robust Invariant Features

The Robust Invariant Features (RIF) algorithm proposed by Lin *et al.* [Lin et al., 2005] starts by searching extrema of the Harris cornerness function (Eq. 2.10) in the scale-space, built by means of Gaussian smoothing. As stated before, the location of an interest point varies slightly with the scale. Let the Local Corner Signature (LCS) of an interest point be its evolution in the scale-space. Several of the LCSs in an image may intersect as the scale increases. The algorithm determines if the intersecting LCSs correspond to the same or to a different local structure, by means of a tracking and grouping process.

Let $S = \{\mathbf{p}_i^l = [x_i^l, y_i^l]^T | l = 1, \ldots, L, i = 1, \ldots, N_l\}$ be a set of multi-scale interest points, where $\mathbf{p}_i^l$ is the interest point i at scale l, L is the total number of scales and N_l is the number of interest points at scale l. The steps of the tracking and grouping process are:

1. Initialize the current level (scale) $l = L$ and the number of groups $k = 0$

2. For the current level l

(a) Assign to a new group G_k each interest point that is not linked to any other interest point in the upper level, and set $k = k + 1$

(b) For each interest point $\mathbf{p}_i^l$ in this level, search for a link to other interest point in the level $l-1$. If the distance to its nearest neighbor in $l-1$ is below a given threshold, that element is assigned to the group of $\mathbf{p}_i^l$. Otherwise, the link ends at that level l.

3. Proceed with the next level $l = l - 1$ and repeat steps 2 and 3 until all interest points are assigned to a group or until the level $l = 0$ is reached.

The result of the algorithm is a set of groups $\{G_1, G_2, \ldots, G_k\}$. These groups are clusters of interest points that correspond the same local structure. The last step is scale adaptation, that is, selecting a representative interest point in each group. Firstly, the algorithm computes the normalized cornerness along the LCSs in the group and the strongest peak is selected. Then, an interpolation gives the exact scale of its representative interest point.

2.1.3 Affine invariance

The main issue of scale-invariant feature extraction methods is that their scale-space representation is based on isotropic Gaussian kernels. They detect isotropic (circular) interest regions that can not cope with certain affine transformations, like those generated by viewpoint variations. This section introduces several state-of-the-art affine-invariant feature detection algorithms. These algorithms extract anisotropic (elliptical) regions by means of either a non-uniform scale-space representation or not. Firstly, we summarize the Harris affine and Hessian affine algorithms, that are an affine extension of the Harris-Laplace and Hessian-Laplace algorithms based on the work of Lindeberg and Gårding [Gårding and Lindeberg, 1996][Lindeberg and Gårding, 1997]. Then, we describe other methods that extract irregular-shaped regions that can be fitted by ellipses.

Harris affine

As in the case of scale invariance, Mikolajczyk and Schmid adapted the Harris interest point detector to affine invariance

[Mikolajczyk and Schmid, 2002][Mikolajczyk and Schmid, 2004b]. Due to the computational burden of building a non-uniform scale-space, the affine-invariant feature detection process starts from an initial set of scale-invariant features. These features are previously extracted using the Harris-Laplace method and then their shape is refined by means of an iterative algorithm inspired by the work of Lindeberg and Gårding [Lindeberg and Gårding, 1997].

The aim of the iterative process is to compute a geometric transformation, known as shape adaptation matrix, for each previously detected scale-invariant feature. Beginning with an initial identity matrix $U^{(0)}$, the algorithm concatenates in each iteration k a new transformation matrix based on the second moment matrix (Eq. 2.65):

$$U^{(k-1)} = (\mu^{-1/2})^{(k-1)} \dots (\mu^{-1/2})^{(1)} U^{(0)} \quad . \tag{2.70}$$

Let $\mathbf{x}^{(0)}$ be the initial position of the processed feature, $\sigma_I^{(0)}$ its initial integration scale and $\sigma_D^{(0)}$ its initial differentiation scale. In each iteration the image is transformed using $U^{(k)}$ and these parameters are updated as follows:

- The integration scale $\sigma_I^{(k)}$ is the one that maximizes the LoG (Eq. 2.68) over scales.
- The derivation scale $\sigma_D^{(k)}$ is computed from the integration scale as $\sigma_D^{(k)} = s\sigma_I^{(k)}$. The parameter s is selected in order to maximize the normalized isotropy $\lambda_{min}/\lambda_{max}$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $\mu^{(k)}$, respectively.
- The spatial location is updated from the transformed point $\mathbf{x}_w^{(k-1)}$, being $\mathbf{x}^{(k-1)} = U^{(k-1)}\mathbf{x}_w^{(k-1)}$. The translation vector $\mathbf{x}_w^{(k)} - \mathbf{x}_w^{(k-1)}$ is computed, being $\mathbf{x}_w^{(k)}$ the point that maximizes the Harris function in the neighborhood of $\mathbf{x}_w^{(k-1)}$. Then, this new point is back-mapped to the original image:

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + U^{(k-1)} \left(\mathbf{x}_w^{(k)} - \mathbf{x}_w^{(k-1)} \right) \quad . \tag{2.71}$$

Mikolajczyk and Schmid proposed two possible termination criteria. Both yield similar results:

- based on μ: the iterative process ends when μ is close to a generic rotation matrix ($\lambda_{min} \approx \lambda_{max}$).
- based on $U = R^T \cdot D \cdot R$: the iterative process ends when the rotation matrix R and the scale matrix D are similar during several iterations.

The iterative process also stops when the divergence between scales in both directions is large, that is, when the difference between the eigenvalues of D is large. In this case, the interest point is rejected. An example of application of the Harris affine feature extraction algorithm is shown in Fig. 2.4.

Hessian affine

The only difference between the Hessian affine feature extractor [Mikolajczyk and Schmid, 2004a] and the Harris affine feature extractor is that the former is based on the Hessian matrix:

$$H(x,y) = \begin{bmatrix} L_{xx}(x,y) & L_{xy}(x,y) \\ L_{xy}(x,y) & L_{yy}(x,y) \end{bmatrix} , \tag{2.72}$$

where L_{xx} and L_{yy} are the second derivatives of L along x and y directions, respectively, and L_{xy} is the mixed derivative. The Hessian interest point detector searches for pixels that maximize both the determinant and the trace of the latter matrix. The multi-scale and the affine-invariant extensions of the Harris interest point detector also apply to this case. We show an example in Fig. 2.4.

Maximally Stable Extremal Regions

The Maximally Stable Extremal Regions algorithm proposed by Matas *et al.* [Matas et al., 2004] has received special attention in recent years. It has proven to be well suited to matching problems [Mikolajczyk et al., 2005b] and as a consequence it has been extensively applied and updated (see, for instance, [Murphy-Chutorian and Trivedi, 2006]. [Donoser and Bischof, 2006a], [Donoser and Bischof, 2006b], [Vedaldi, 2007] and [Forssén, 2007]). The MSER algorithm also inspired a new shape-adapted SIFT descriptor [Forssén and Lowe, 2007].

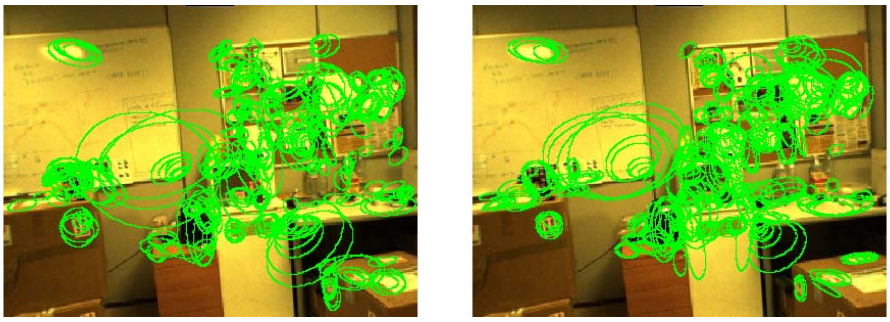


Figure 2.4: Example of application of the Harris affine and Hessian affine algorithms to the image in Fig. 2.1. Left: Harris affine results. Right: Hessian affine results.

A Maximally Stable Extremal Region (MSER) is a connected component in a thresholded image. Let's first define the concept of maximal and minimal regions. Given an image and an intensity threshold t, all pixels for which intensity is below t are labeled as black pixels, and those for which intensity is above t are labeled as white pixels. If $t = 0$ we have a completely white image. As we increase t, small black regions start to appear. Further increments of t result in the growing of these regions, that eventually will merge. When the maximum value of t is reached, the image will be completely black. The maximal regions are all those connected components found during this thresholding process. If the intensity of the image is inverted and we repeat the thresholding process, what we get are minimal regions.

A maximal or minimal region is extremal if the intensity of every pixel in that region is higher or lower than the intensity of every pixel in its contour. A maximal or minimal extremal region is stable if its size remains constant during several iterations of the thresholding process. The regions detected by the algorithm are invariant to intensity variations and also to scale and affine transformations. Another advantage of this algorithm is that its computational cost is low.

The MSER algorithm works as follows. Firstly, the image pixels are ordered in ascending or descending intensity order. Following this order, the pixels are stored into an union-find structure, that represents a list of connected components and their intensity values. This step computes the area of any connected component as a function of intensity. Finally, those

intensity levels that produce local minima of the rate of change of the area function are selected as thresholds in order to obtain the MSERs. The output of the algorithm is a set of MSERs represented by its location (a seed pixel) and an intensity threshold. The MSERs have irregular shapes, but they are usually transformed into ellipses (see Fig. 2.5), that is, anisotropic regions that correspond to affine invariant features.

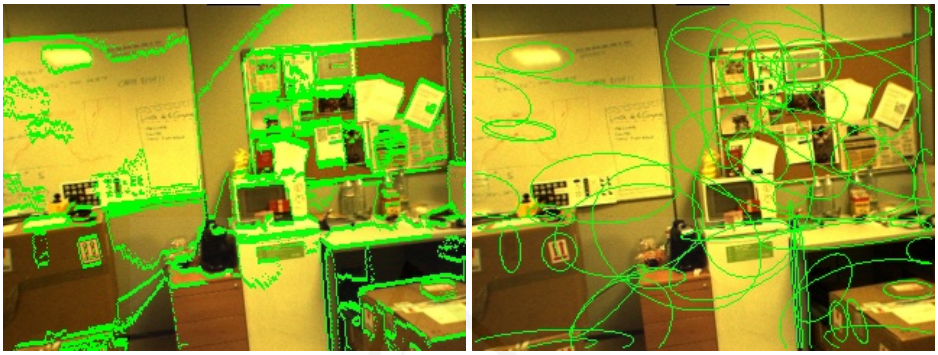


Figure 2.5: Example of application of the Maximally Stable Extremal Regions algorithm to the image in Fig. 2.1. Left: extracted regions. Right: the regions are transformed into ellipses.

Maximally Stable Corner Clusters

The Maximally Stable Corner Clusters (MSCC) algorithm proposed by Fraundorfer *et al.* [Fraundorfer et al., 2005] is strongly inspired in the MSER method. The main difference with respect to other feature extraction algorithms is that each interest region is built from a local set of interest points. Each region represents an interest point constellation. Fraundorfer *et al.* state that the robustness of these regions to changes in viewpoint is higher than that of individual interest points. Only those interest point clusters that satisfy a stability criterion are labeled as interest regions.

The first step of the algorithm is the extraction of interest points by means of the computation the structure tensor [Bigun and Granlund, 1987] and the Harris cornerness (see Eq. 2.10) for each pixel in the image. A large amount of corners is selected; specifically, those that are a local maxima over a given noise level. Thus, the results of the algorithm do not depend on a predefined

cornerness threshold. The selected corners are the basis of the next step, in which each corner represents a node in a 2D weighted Minimal Spanning Tree (MST)[2]. The weights of the tree edges are the distances between their connected nodes.

The second step of MSCC is based on a multi-scale clustering process. The MST is split using a range of weight thresholds. For each threshold t, all those edges for which their weight is above t are pruned. Each subtree corresponds to a interest point cluster that in turn represents an image region. Like in the case of the MSER algorithm, the MSCC method searches for all those clusters that are stable in a wide range of values of t. The detected regions also have irregular shapes and can be transformed into ellipses. An example of application is shown in Fig. 2.6.

MSCC is an interesting technique that focuses on clearly different image features when compared to other feature extractors. However, it has not as high performance as that of MSER, Harris affine or Hessian affine [Fraundorfer et al., 2005].

Figure 2.6: Example of application of the MSCC algorithm to the image in Fig. 2.1. Left: the MST spanning all the detected Harris interest points (represented as circles). Right: Detected regions.

Another approach that achieves affine invariance by means of interest point grouping is the one introduced by Brown and Lowe [Brown and Lowe, 2002]. In this case, interest points are detected as the maxima of the DoG function, and a descriptor is assigned to each

[2] for a definition of Minimal Spanning Tree see Section 4.2.

interest point group, being this descriptor invariant to affine transformations.

Edge Based Regions

Edge Based Regions and Intensity Based Regions are two algorithms proposed by Tuytelaars *et al.* [Tuytelaars and Gool, 2004]. Both methods are based on the selection of seed points from which the algorithm searches affine-invariant regions. Focusing on these seed points decrease the computational burden of these methods. The first method, Edge Based Regions (EBR), relies on the Harris corner and the Canny edge detection algorithms [Canny, 1986].

Let $\mathbf{p} = (x_p, y_p)$ be a Harris corner detected on an edge. Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be two points that move away from $\mathbf{p}$ along both directions of the edge. Their velocity is matched by means of the equality of two invariant parameters l_1 and l_2, calculated as

$$l_i = \int abs(det(\mathbf{p}_i^{(1)}(s_i)\mathbf{p} - \mathbf{p}_i(s_i)))ds_i, i = 1, 2 \ , \tag{2.73}$$

where s_i is the parameter of the curve defined by the edge, abs the absolute value, and $\mathbf{p}_i^{(1)}(s_i)$ the first derivative of $\mathbf{p}_i(s_i)$ with respect to s_i. When $l_1 = l_2$ then we name this value l. For each l value, the points $\mathbf{p}_1(l)$ and $\mathbf{p}_2(l)$, together with $\mathbf{p}$, define a parallelogram with an area Ω as a function of l. Points $\mathbf{p}_1$ and $\mathbf{p}_2$ stop and a new affine region is detected when any photometric quality of Ω reaches an extremal value. These photometric qualities could be, for instance, functions based on moment invariants. The previous method can not be applied in the case of straight edges, due to the fact that $l = 0$ along all the edge. In this case, the algorithm searches the intersection of minima of two photometric functions.

The detected regions are parallelograms but these regions are usually transformed into ellipses. The detected parallelograms are not centered in their seed point; planar regions are detected from corners and the algorithm is more robust to false positives (image background). An example of application is shown in Fig. 2.7

Intensity Based Regions

The second feature extractor proposed by Tuytelaars *et al.* is the Intensity Based Regions (IBR) algorithm [Tuytelaars and Gool, 2004]. Their experimental results demonstrated that EBR and IBR are complementary; that is, that most of the regions detected by EBR are not detected by IBR, and vice versa. Intensity Based Regions are based solely on intensity; neither corner points nor edges are required. The seed points are local maxima of image intensity.

The algorithm analyzes the intensity along straight rays that emanate from each seed point by means of the following equation:

$$f_I(t) = \frac{abs(I(t) - I_0)}{max(\frac{\int_0^t abs(I(t)-I_0)dt}{t}, d)} \quad , \tag{2.74}$$

where t is the Euclidean archlenght along the ray, $I(t)$ is the intensity on t, I_0 is the intensity of the seed, and d is a small number that is added in order to avoid divisions by zero. The algorithm locates the maximum $f_I(t)$ along each ray, that is produced at the edges of homogeneous regions. The maxima along all rays are linked together in order to build an affine region. This irregular shape is then transformed into an ellipse. The variability of intensities in the ellipse is higher, thus increasing its distinguishability. However, the ellipse loses some of the planarity of the detected region.

If several maxima of $f_I(t)$ are found along a ray, instead of selecting the global maxima the algorithm imposes a continuity constraint: given a maximum in a ray, the nearest local extrema in adjacent rays are selected. By the other hand, the Intensity Based Regions are no centered on the seed points. Slight variations of the seed location produce the same Intensity Based Region, increasing matching robustness. An example of application is shown in Fig. 2.7.

2.2 Feature extraction based on local visual saliency

In this section we will briefly define the term local visual saliency and we will also introduce the Scale Saliency algorithm proposed by Kadir and

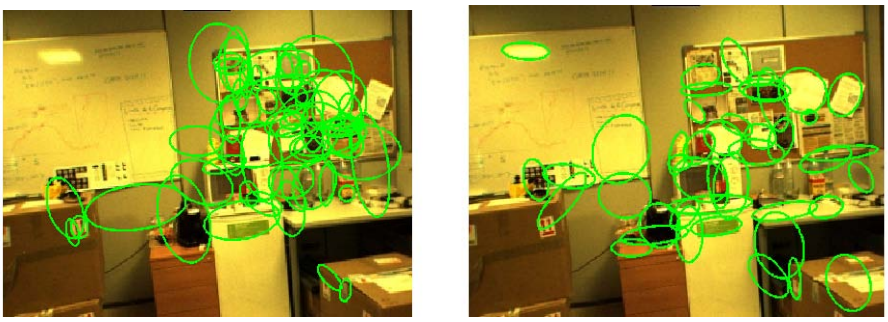


Figure 2.7: Example of application of the algorithms proposed by Tuytelaars *et al.* to the image in Fig. 2.1. Left: edge based regions (EBR). Right: intensity based regions (IBR).

Brady [Kadir and Brady, 2001]. All our contributions in this Thesis rely on these two elements.

The word saliency refers to the quality of being salient. A given entity is salient if it stands out from other ones in the same domain. The definition can be directly applied to image analysis. An image region is visually salient if it is distinguishable from the rest of elements in the image, in terms of intensity, orientation or any other property. See, for instance, Fig. 2.8. In the left image, the intensity of the green circle makes it salient with respect to the rest of circles. In the case in the right image, the salient element is the one with different orientation. Both elements may be considered globally (because they are salient with respect to the rest of elements of the image) and locally (because they are salient with respect to their neighborhood) salient.

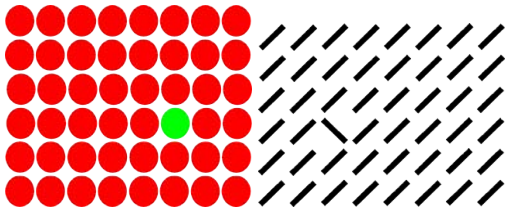


Figure 2.8: Two examples of visual saliency.

In order to extract useful features from images, our algorithms will be focused on the detection of locally salient regions (in terms of intensity), that is, regions with a locally distinguishable intensity distribution. These

regions are supposed to remain salient under several transformations, like for instance scale variations or affine distortion [Kadir et al., 2004]. This property is required by high-level vision tasks that rely on this kind of features.

2.2.1 Gilles image saliency

Gilles was the first author to relate saliency in the context of visual image feature extraction to Information Theory [Gilles, 1998]. Indeed, Shannon's entropy seems to be an adequate tool to estimate image saliency. Shannon's entropy is a measure of the unpredictability associated with a random variable. It basically refers to the amount of information contained in a message. The less predictable the value of a random variable is, the more information it provides. Gilles defined salient regions in grayscale images as piecewise regions that were locally unpredictable, that is, the set of highest-informative regions in an image (high-entropy regions). Given a pixel $\mathbf{x}$, which intensity lies in the domain $D = \{d_1, \ldots, d_n\}$ (between 0 and 255 in the case of grayscale images), and its local neighborhood $R_{\mathbf{x}}$, its saliency is estimated by means of Shannon's entropy [Cover and Thomas, 1991]

$$H_{D,R_{\mathbf{x}}} = -\sum_i P_{D,R_{\mathbf{x}}}(d_i) \log_2 P_{D,R_{\mathbf{x}}}(d_i) \ , \tag{2.75}$$

where $P_{D,R_{\mathbf{x}}}(d_i)$ is the proportion of pixels in R_x which intensity is d_i. Low entropy values correspond to predictable or low informative random variables, that is, random variables in which the probability of a given random value is much higher that of the rest of values. On the contrary, higher entropy values correspond to unpredictable random variables, in which the probability of all their possible random values is similar. Seen from the perspective of Information Theory, it is obvious that a feature extraction algorithm should detect the highest-entropy regions of the image. Fig. 2.9 shows an example of how this measure characterizes salient and not salient regions. Note that predictability in homogeneous regions is high, and as consequence, these regions are low-salient. Our contributions in Chapter 3 are based on this fact.

The feature extraction algorithm proposed by Gilles works as follows: firstly, the size and shape of $R_{\mathbf{x}}$ is set. We will assume square regions with

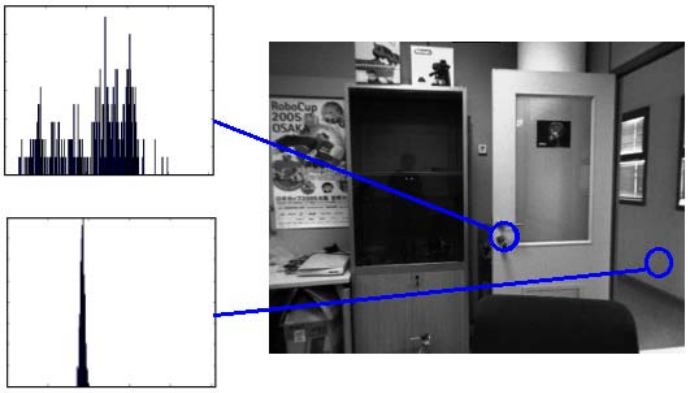


Figure 2.9: Saliency by means of Shannon's entropy. We build a histogram to approximate the intensity probability density function (pdf) of each region. Then, entropy is estimated from this histogram. Homogeneous regions will tend to produce peaked histograms. These are regions are less informative, and as consequence less salient, than regions with higher intensity unpredictability.

a fixed radius measured in pixels. This radius is called scale. Then entropy is estimated for each pixel using Eq. 2.75. All pixels for which entropy is below a given threshold are discarded. The result is a binarized image containing blobs. Finally, the algorithm selects the local entropy maxima in this blobs. These points correspond to the most salient features in the image. Fig. 2.10 shows an example of application. In this example the scale was set to 7 and the entropy threshold was set to 6.6. Most of the salient features in Fig. 2.10 were detected on cars, due to the fact that these cars locally stand out from the road.

Although entropy-based saliency estimation is intuitive and simple, this algorithm suffers some limitations [Kadir and Brady, 2001]. The most evident one is the fixed scale constraint: due to the fact that scale is a preset parameter, the saliency search is constrained to a narrow range of scales. The cars image in Fig. 2.10 is a clear example. The pedestrian in the top right part of the image is salient with respect to the pavement; however, the scale of that feature is smaller, and as a consequence, it is not labeled as a salient feature. The algorithm is also very sensitive to small noise. Finally, highly textured regions

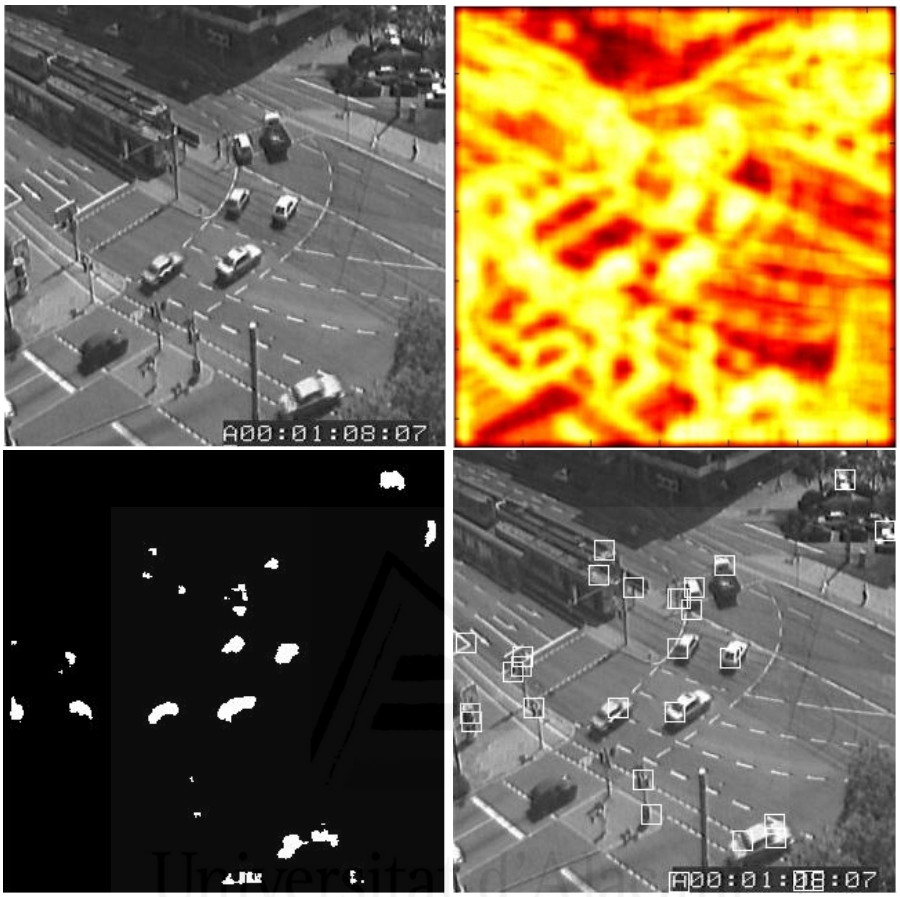


Figure 2.10: Gilles algorithm results. Top left: the input image. Top right: image saliency. Brighter intensity represents higher saliency. Bottom left: salient pixels after thresholding. Bottom right: final salient features.

with high intensity variations are usually labeled as salient features, even if these regions are part of a larger textured region and are not salient from a perceptual point of view.

2.2.2 Scale Saliency

Kadir and Brady proposed their Scale Saliency algorithm in order to deal with the limitations of the Gilles algorithm. This algorithm detects salient regions not only in the image-space but also in the scale-space. Scale limitation is solved by applying the entropy measure to each pixel while

isotropically increasing the size of $R_{\mathbf{x}}$. The output of the algorithm is a set of circular salient regions of different size. Although Kadir and Brady also proposed an anisotropic Scale Saliency algorithm that can cope with affine transformations [Kadir et al., 2004], throughout this thesis we focus on the original isotropic method. The application of the scale-space representation requires the redefinition of the term visual saliency, now subject to two possible interpretations: a salient region may be a region that is salient either in a wide or a narrow range of scales. Kadir and Brady adopted the second definition of salient region. They stated that saliency in a wide range of scales is a consequence of fractal or random images, or self-similarity. In the Scale Saliency algorithm a salient region must be distinguishable both in space and scale space.

The algorithm works as follows. Firstly, the range of scales is set between a minimum scale s_{min} and a maximum scale s_{max}. Then, the entropy of each pixel $\mathbf{x}$ at each scale s is estimated from its intensity pdf (see Fig. 2.11):

$$H_D(s, \mathbf{x}) = -\sum_{d \in D} P_{d,s,\mathbf{x}} \log_2 P_{d,s,\mathbf{x}} \quad . \tag{2.76}$$

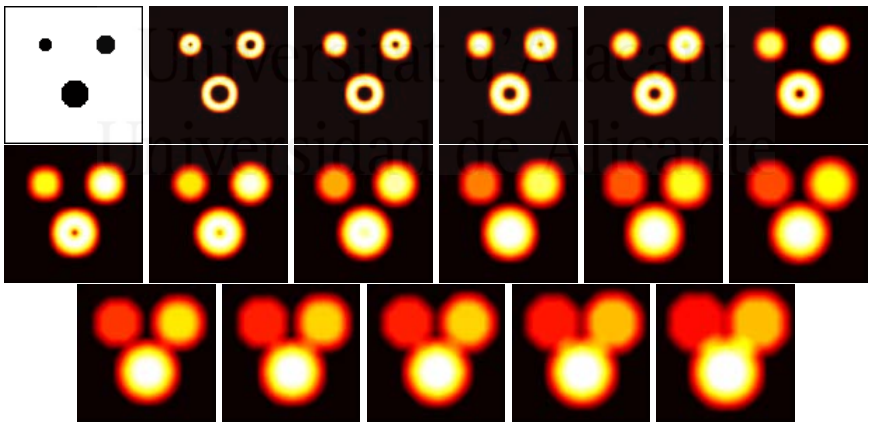


Figure 2.11: From left to right and from top to bottom: an example synthetic image and its estimated entropy in the range of scales between $s_{min} = 5$ and $s_{max} = 20$. Brighter intensities represent higher entropy values.

Next, the algorithm detects entropy peaks in the scale-space, that is, local maxima of the entropy function:

$$S_p = \{s : H_D(s-1, \mathbf{x}) < H_D(s, \mathbf{x}) > H_D(s+1, \mathbf{x})\} \ . \tag{2.77}$$

The entropy of $\mathbf{x}$ in scales $s \in S_p$ is weighted by means of a measure of self-dissimilarity in the scale-space. The self-dissimilarity measure allows the direct saliency value comparison between pixels at different scales and penalizes those features that are salient in a wide range of scales (see Fig. 2.12 and Fig. 2.13):

$$W_D(s, \mathbf{x}) = \frac{s^2}{2s-1} \sum_{d \in D} |P_{d,s,\mathbf{x}} - P_{d,s-1,\mathbf{x}}| \ . \tag{2.78}$$

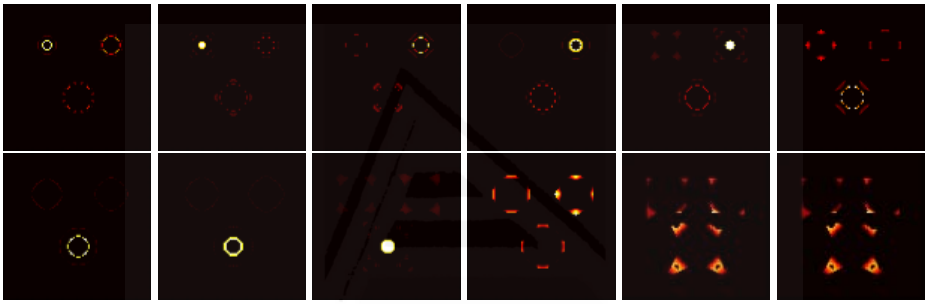


Figure 2.12: Weighted entropy in the range of scales between $s_{min} = 6$ and $s_{max} = 17$. Only weighted entropy peaks are shown. The results were extracted from the synthetic image in Fig. 2.11. Brighter intensities represent higher weighted entropy values. We can see the high saliency of the circle centers in scales 7, 10 and 14. These are the actual circle scales.

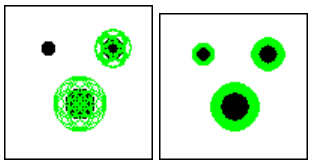


Figure 2.13: Output of the Scale Saliency algorithm for the synthetic image in Fig. 2.11, without (left) and with (right) entropy weighting (Eq. 2.78). The 0.05% and 0.02% most salient features are shown, respectively.

The saliency of a region is its weighted entropy. Saliency is computed only for those scales $s_p \in S_p$:

$$Y_D(s_p, \mathbf{x}) = H_D(s_p, \mathbf{x}) W_D(s_p, \mathbf{x}) \ . \tag{2.79}$$

The output of the Kadir and Brady algorithm is an array $Y(S, X)$ that stores the saliency for every pixel $\mathbf{x} \in X$ in the selected scales. The salient features of the image are the maxima in $Y(S, X)$. Fig. 2.13 shows an example of application to a synthetic image. In Fig. 2.15 we can see examples of salient features extracted from other images, including the cars image previously shown in Fig. 2.10. Now the algorithm correctly labels the pedestrian and even the trash bin next to him/her as salient features.

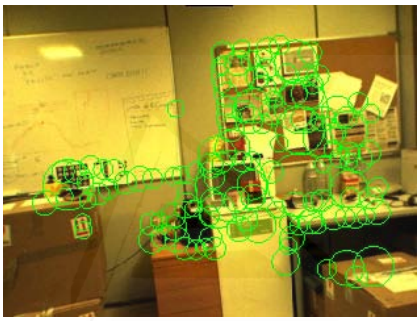


Figure 2.14: Example of application of the Scale Saliency algorithm to the image in Fig. 2.1.

Local non-maximum suppression, or salient feature clustering, is an additional and useful post-processing step. Its aim is to merge the information of salient features that are close in the image-space. This step decreases the amount of extracted salient features, decreasing the temporal and spatial complexity of the applications that rely on these features. It also increases robustness [Kadir and Brady, 2001]. Being K and V_{th} two preset parameters, the steps of the feature clustering process are:

- Choose the highest salient region in $Y(S, X)$
- Find its k nearest neighbors
- Calculate the variance V of their center points, the mean scale s_{mean} and the mean location $\mathbf{x}_{mean}$.

- Find distance D in $\mathcal{R}^3$ (image row, image column, scale) from the selected region to salient regions already clustered
- Create a new salient region with scale s_{mean} and location $\mathbf{x}_{mean}$ if $D > s_{mean}$ and $V < V_{th}$.
- Repeat from the second step with the next highest salient region until a percentage of elements in $Y(S, X)$ is processed.

An example of application of the Kadir and Brady algorithm, including the feature clustering post-processing step, to the image in Fig. 2.1 is shown in Fig. 2.14. Note that in this case the extracted regions are isotropic, and not affine regions (the affine-invariant Scale Saliency algorithm is described in [Kadir et al., 2004]).

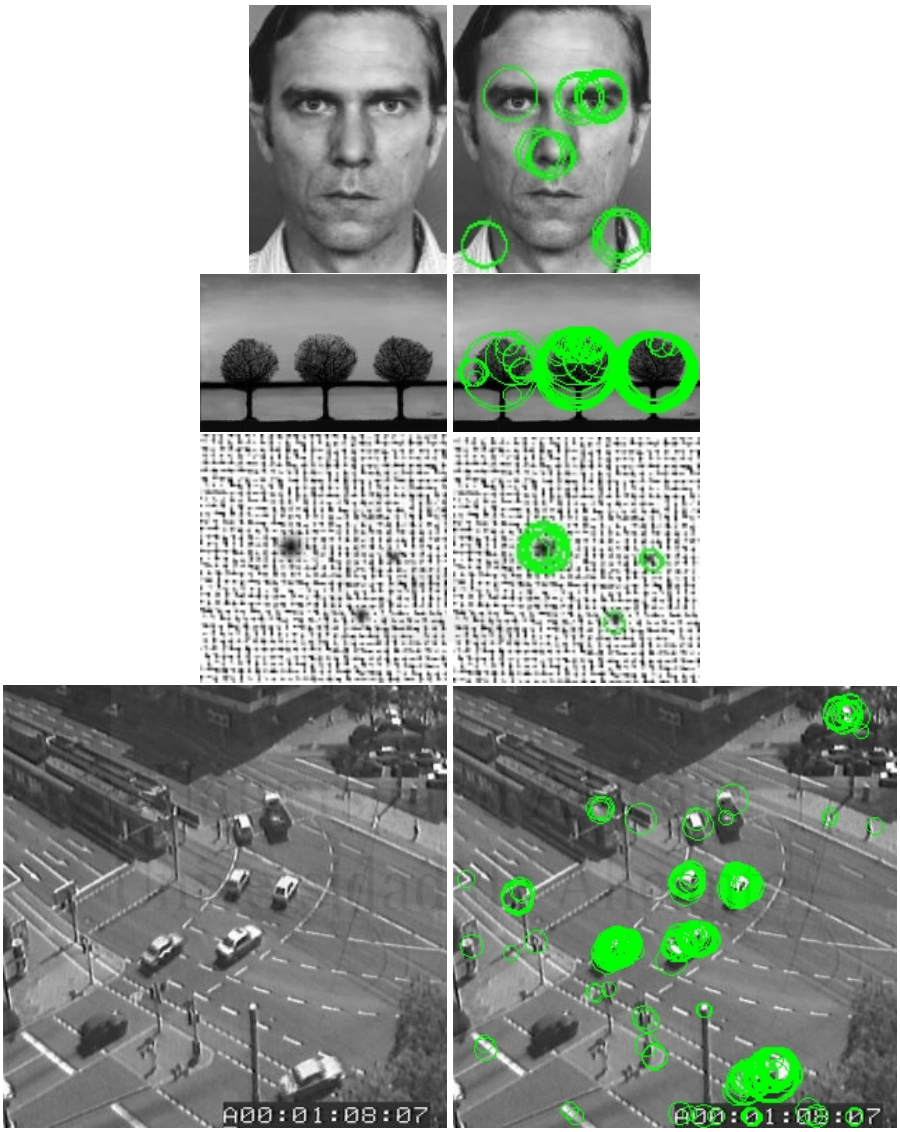


Figure 2.15: Results of the Scale Saliency algorithm for several input images. In order to obtain an optimal output, the parameters of the algorithm (the range of scales and the amount of displayed salient features) were set independently for each image.

Chapter 3

Bayesian filtering of Scale Saliency

3.1 Introduction

In this chapter we address the main problem of the Scale Saliency algorithm: its required computation time is remarkably higher than those of the other state-of-the-art visual feature extraction methods [Mikolajczyk et al., 2005b]. Even if histogram are reused along the scale-space (see Section 3.6.3), the bottleneck of the algorithm is the estimation of Shannon's entropy (Eq. 2.76) for every pixel at every scale in the range $[s_{min}, s_{max}]$.

This fact motivated our study of the evolution of the entropy function in the image and scale-spaces, in order to demonstrate a simple but intuitive hypothesis: image regions that are not salient (i.e. homogeneous) at higher scales are also probably not salient in the rest of the scale-space. The idea in this chapter is to take advantage of entropy properties in order to filter non-interesting points, that is, points that probably will not be part of the output of the Scale Saliency algorithm, before applying the method proposed by Kadir and Brady to the rest of the image. In this chapter we will apply Information Theory measures in order to assess the applicability of our algorithm to a given set of images and to estimate the error probability, that is, the probability that actual salient features are filtered

Firstly, in Section 3.2, we show the initial results of our study of

the evolution of the entropy function in the image and space-scales. We also show how statistical inference supports our hypothesis and as a consequence we present a filtering first approach in Section 3.3. However, this approach is not feasible: the value of its only parameter can only be estimated *a posteriori*, that is, after the Scale Saliency algorithm has been applied to the image. Furthermore, this parameter value is only valid for that image; different images yield different parameter values. Starting from our first filtering method, and based on previous statistical edge and contour detection approaches [Konishi et al., 2003][Cazorla et al., 2002], in Section 3.4 we present the fundamentals of our final filtering algorithm, that is summarized in Section 3.5. In our Bayesian filtering algorithm, the statistics of a set of images belonging to a same environment or category allows the *a priori* estimation of a filtering threshold applicable to new input images. Information Theory is used here to validate image groups and to estimate the probability of error. We show several experimental results in Section 3.6. Finally, in Section 3.7 we show a practical application of our algorithm to the field of robot localization.

3.2 Analysis of entropy in image and scale-space

In this section we will summarize the conclusions that led us to the hypothesis developed throughout this chapter: *image regions that are homogeneous or low-salient at the highest scale will probably also be homogeneous or low-salient at lower scales*. These conclusions were extracted from the observation of the evolution of the entropy function in the image and scale-space and from statistical inference. In Fig. 3.1, for instance, we can see a 3D representation of the saliency of the cars image (see Fig. 2.10) for four different scales. The saliency function is continuous. Its evolution in the image-space is smooth, even smoother in the case of higher scales.

This fact motivated us to test the smoothness of the entropy function in the scale-space. The aim of this test was to obtain enough empirical evidence in order to start a deeper analysis that confirmed our hypothesis. We represented the evolution of the entropy function in the scale-space for several input images. An example is shown in Fig. 3.2. We would need a 4D

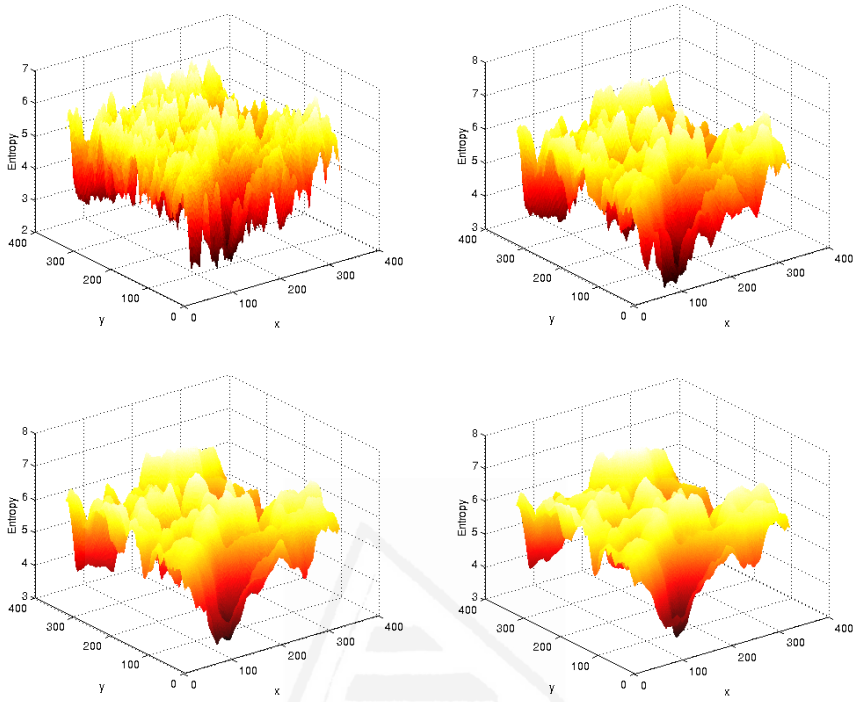


Figure 3.1: 3D representation of the entropy of the cars image at scales 8, 12, 16 and 20.

plot in order to represent the evolution both in image and scale-space. Thus, we chose a simpler representation. Specifically, in Fig. 3.2 we are showing the evolution of the entropy function in the scale-space (in the range between $s_{min} = 5$ and $s_{max} = 15$) for two different rows of the cars image. As can be seen, the entropy function changes smoothly in scale-space for any pixel. This evolution of entropy was also observed in a large set of images. The consequence is that less-salient pixels at s_{max} are also less-salient at s_{min}. Further empirical evidence is given in Fig. 3.3, where salient regions at scale 5 are superimposed on salient regions at scale 20. As can be seen, the correspondence is high.

After all these preliminary tests, we conducted several experiments aimed to prove the relationship between the most salient features in an image and the values of the entropy function at s_{min} and s_{max}. Specifically, our hypothesis was:

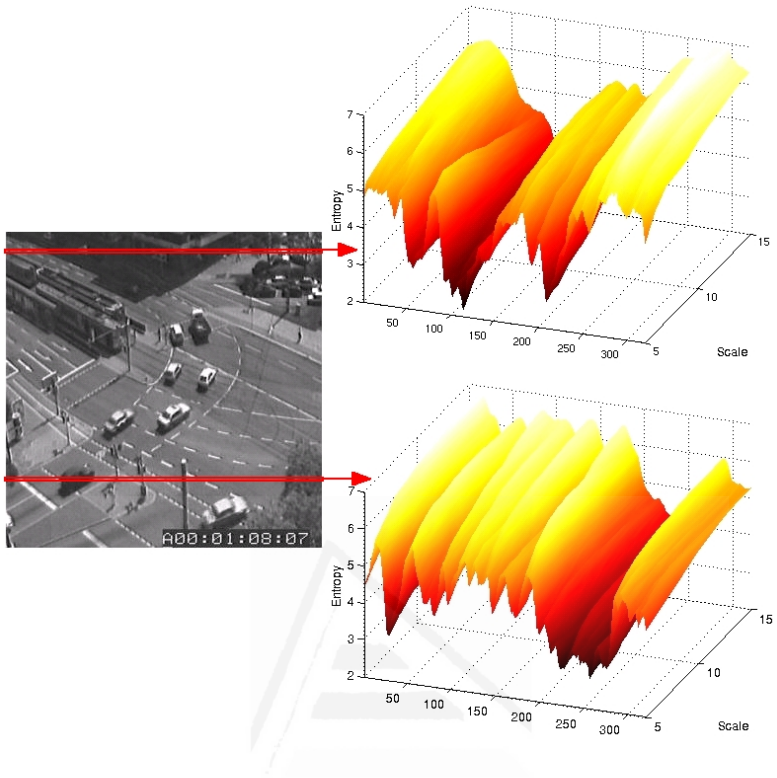


Figure 3.2: Evolution of the entropy function in the scale-space for two rows of the cars image.

- If the entropy of a pixel at s_{min} **and** at s_{max} is high, it is likely to be part of the most salient features of the image.
- If the entropy of a pixel at s_{min} **or** at s_{max} is high, it is also likely to be part of the most salient features of the image, but with lower probability
- Pixels with a low entropy value at s_{min} and at s_{max} are not likely to be part of the most salient features of the image.

We used the *Object categories* dataset[1] published by the *Visual Geometry Group* (University of Oxford). This dataset is composed of 12 image categories. Each category contains from 126 to 1155 images of variable size (see Fig. 3.4). We selected 1000 random pixels from 240 randomly chosen

[1]http://www.robots.ox.ac.uk/~vgg/

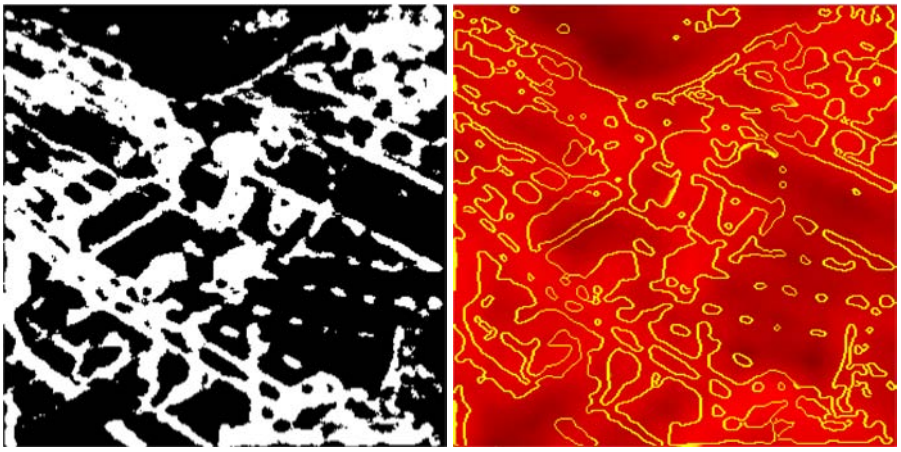


Figure 3.3: Correspondence of salient regions in the cars image at different scales. Left: represented in white color, pixels for which entropy at scale 5 is above a given threshold. Right: represented in yellow color, superimposition of the boundaries of the connected components in left image on a representation of the saliency at scale 20, where brighter intensities correspond to higher saliency values.

images (20 from each category), making a total of 240000 points. Then, using Eq. 2.76 we computed their entropy in the range of scales from $s_{min} = 5$ to $s_{max} = 20$. In order to prove our hypothesis, we defined a set of variables:

- f_3: ratio between h_{min} and H_{min}
- f_5: ratio between h_{max} and H_{max}
- f_7: ratio between h^* and H_{min}
- f_9: ratio between h^* and H_{max}

where, for a given pixel, h_{min} is its entropy value at s_{min}, h_{max} is its entropy value at s_{max} and h^* is its maximum entropy value in the scale-space. By the other hand, H_{min} is the highest entropy at s_{min} in the image and H_{max} is the highest entropy at s_{max} in the image. The standard correlation between these variables is shown in Table 3.1. We focus our analysis on the correlations highlighted in the table. As can be seen, correlation is strong (next to 1) in all these cases. These correlation values indicate a strong lineal dependency

between each pair of variables, and suggest that the entropy of a point at s_{min} and s_{max} could help to locate salient pixels. Moreover, correlation values of f_5 are higher than those of f_3. Thus, h_{max} seems to be more informative than h_{min}.

Table 3.1: Linear correlation between the analyzed variables.

	f_3	f_5	f_7	f_9
f_3	1	0.8153	**0.8326**	**0.8278**
f_5	0.8153	1	**0.9917**	**0.9960**
f_7	**0.8326**	**0.9917**	1	0.9942
f_9	**0.8278**	**0.9969**	0.9942	1

Fig. 3.5 shows the relationship between the pairs of variables highlighted in Table 3.1, based on the values of the selected points selected from the *Visual Geometry Group* dataset. The value of f_3 and f_5 put a lower bound on the maximum entropy of a pixel in the scale-space. The probability that a pixel is part of the most salient features is higher if its entropy is high at s_{max} or s_{min}. However, linear regression (the red line in Fig. 3.5) does not provide any useful information.

Another measure that supports the idea of a strong mutual dependence between these variables is multiple correlation. The degree of dependence of a variable y with respect to two independent variables x_1 and x_2 is given by:

$$R = \sqrt{\frac{r_{y,x_1}^2 + r_{y,x_2}^2 - 2r_{y,x_1}r_{y,x_2}r_{x_1,x_2}}{1 - r_{x_1,x_2}^2}}\ , \tag{3.1}$$

where $r_{a,b}$ is the correlation factor between the variables a and b. In our case, and taking f_3 and f_5 as independent variables, R is 0.9926 and 0.9964 for f_7 and f_9, respectively. This result suggests that this dependence between variables is stronger than the linear one. Fig. 3.6 shows a 3D representation of the relationships between the analyzed variables. Both f_3 and f_5 put a lower bound similar to the one shown in Fig. 3.5.

The lower bound in Fig. 3.6 (bottom) forms a plane. The equation of that plane approximates the minimum expected value of the maximum saliency of any pixel in the scale-space. It may provide a formal test aimed to discard image pixels that probably do not belong to the set of the most salient

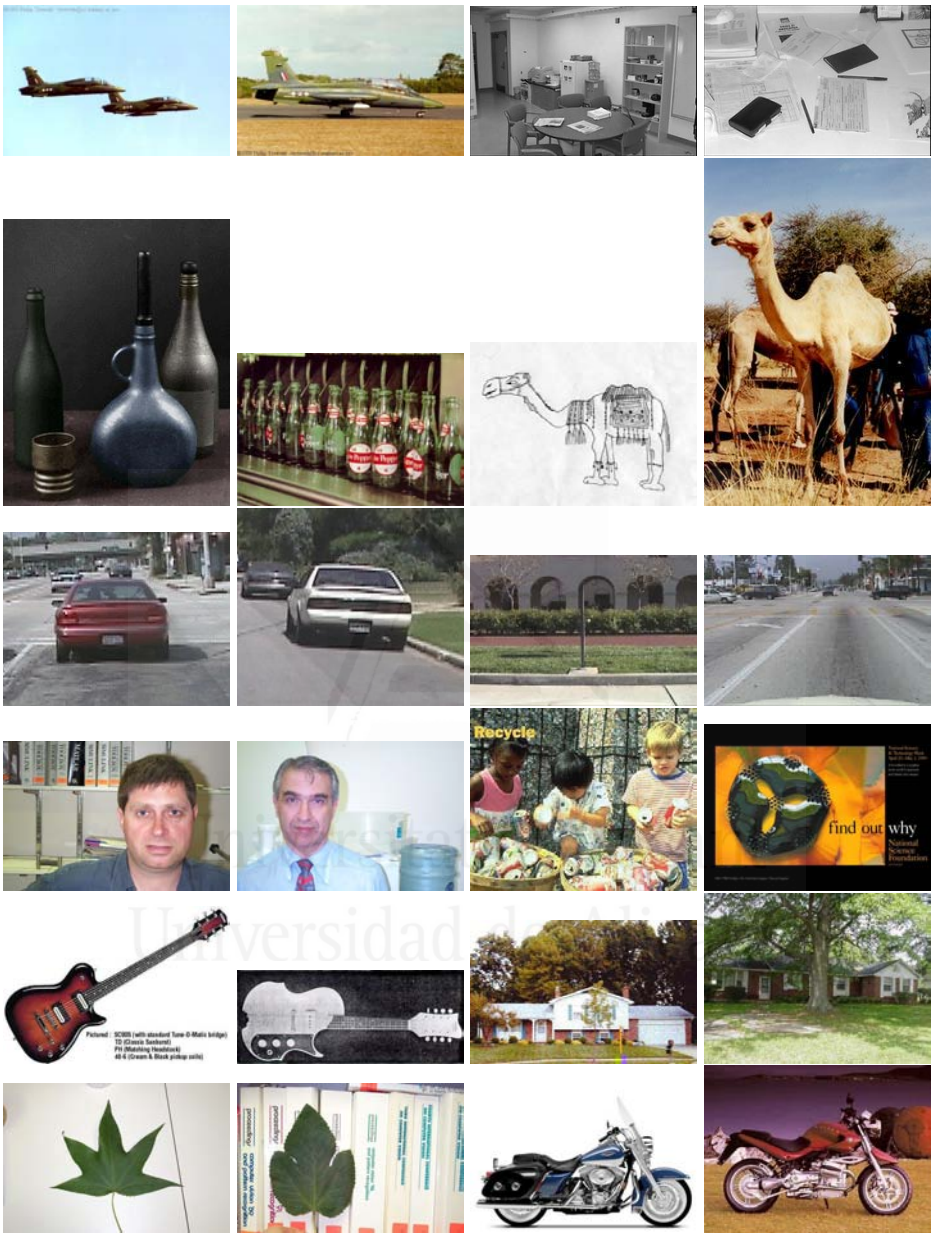


Figure 3.4: Two example images from each category in the *Object categories* dataset. From left to right and from top to bottom: *airplanes_side*, *background*, *bottles*, *camel*, *cars_brad*, *cars_brad_bg*, *faces*, *google_things*, *guitars*, *houses*, *leaves* and *motorbikes_side*.

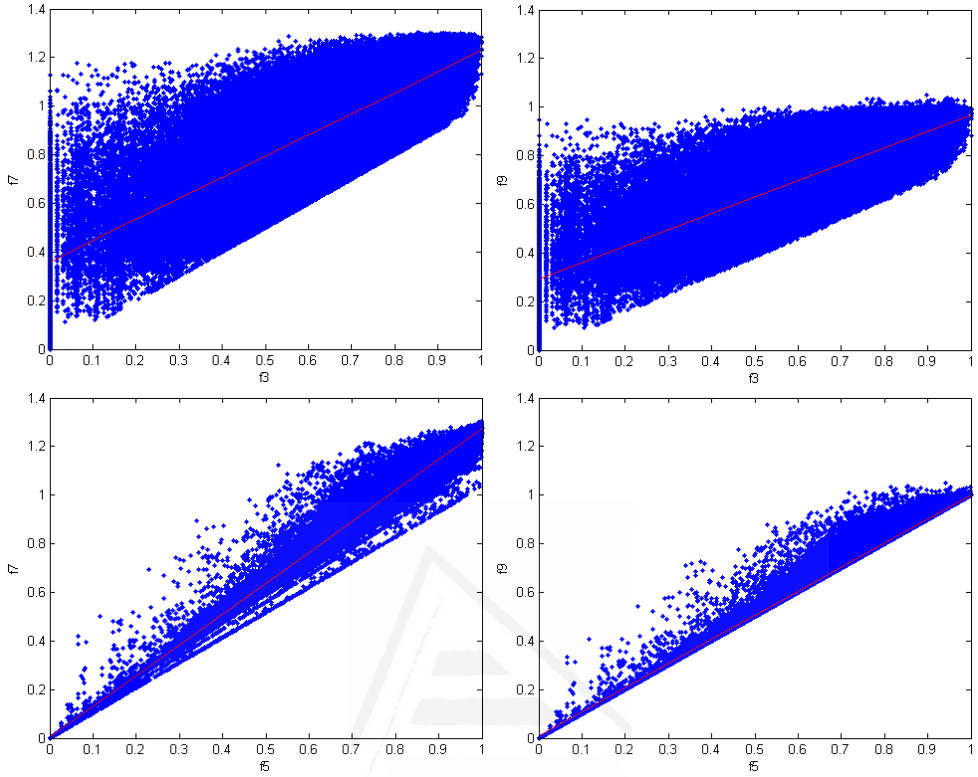


Figure 3.5: Relationship between the analyzed variables. The red line represents the linear regression between each pair of variables.

features in that image. As a consequence, a complete exploration of the scale-space would not be required during the Scale Saliency algorithm, and its time complexity would decrease. We applied the 3D Hough transform method [Hough, 1962][Duda and Hart, 1972][Vosselman and Dijkman, 2001] in order to estimate the parameters of the plane. Given the equation of the plane in $\mathcal{R}^3$

$$f_9 = s_x f_3 + s_y f_5 + d \ , \tag{3.2}$$

the parameters to be estimated are the slope of the plane in the x direction (s_x), the slope of the plane in the y direction (s_y) and the height from the origin in the z direction (d). The Hough Transform method involves the computation of a 3D array of votes, in which each cell corresponds to a

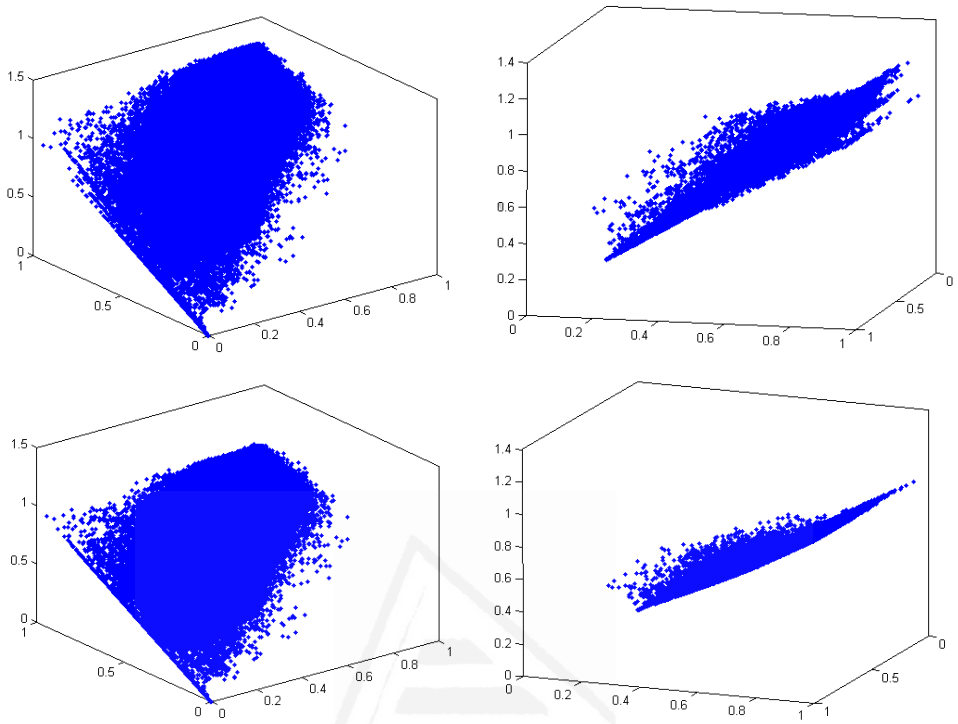


Figure 3.6: Top: relationship between the dependent variable f_7 (vertical axis) and the independent variables f_3 and f_5 . Bottom: relationship between the dependent variable f_9 (vertical axis) and the independent variables f_3 and f_5.

discrete subset of the $\{s_x, s_y, d\}$ space. Any point in the lower bound plane votes for the array cells that represent all the possible planes that pass through that point. We constrained the search-space to the parameter values close to those of the expected plane in order to decrease time and space requirements. Finally, the most voted cell gives the parameters of the lower bound plane. This plane is shown in Fig. 3.7.

The parameters of the extracted plane were $s_x = 0$, $s_y = 1.01$ and $d = 0.015$. Our conclusion is that the value of h^* strongly depends on h_{min} and h_{max}, but the lower bound of h^* depends only on h_{max}. In the context of the Scale Saliency algorithm we conclude that the probability that a pixel is part of the most salient image features is strongly determined by its entropy at s_{max}. This conclusion supports the hypothesis that homogeneous

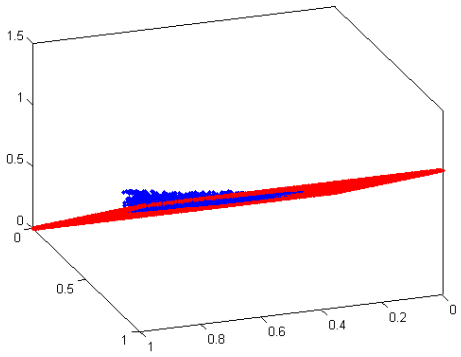


Figure 3.7: Planar approximation obtained by 3D Hough transform of the lower bound of f_9 put by f_3 and f_5.

or non-salient image regions at higher scales tend to also be homogeneous or non-salient at lower scales. These regions are unlikely to contain any salient feature.

3.3 A first approach to pixel filtering before Scale Saliency application

In the previous section we have demonstrated that the entropy of a point at s_{max}, given a range of scales $[s_{min}, s_{max}]$, is enough to put a bound on the probability that a pixel is part of the most salient features in the image. Given H_{max} (the maximum entropy at s_{max} for any pixel in the image), all those points for which their entropy h_{max} at s_{max} is close to H_{max} ($h_{max}/H_{max} \approx 1$) are likely to be part of the these salient regions. Based on these results in this section we present a first simple algorithm to decrease the computational burden of the Scale Saliency algorithm. The main idea is to set a threshold σ so that all pixels for which $h_{max}/H_{max} < \sigma$ will be filtered before applying the Kadir and Brady method. Entropy at s_{max} is normalized with respect to H_{max}. In this way we may set a general threshold σ that could be applied to several images (see Section 3.4). The steps of the filtering algorithm are:

1. Calculate the local entropy H_D at scale s_{max} for each pixel x
2. Set a normalized entropy threshold $\sigma \in [0, 1]$

3. $X = \left\{ \mathbf{x} : \frac{H_D(\mathbf{x}, s_{max})}{\max_{\mathbf{x}}\{H_D(\mathbf{x}, s_{max})\}} > \sigma \right\}$

4. Apply the Scale Saliency algorithm only to $\mathbf{x} \in X$

The threshold σ is a critical parameter. if σ is too high, too many pixels will be discarded and several salient features could be removed. If σ is too low, few pixels are discarded and as a consequence the *total execution time*[2] may not decrease, or even increase. An example of application of the filtering algorithm to the cars image is shown in Fig. 3.8. As σ increases, more points are discarded, decreasing the total execution time. In Fig. 3.8 (right) the value of σ was too high; thus, the Kadir and Brady algorithm produced false negatives and positives. Fig. 3.9 gives an insight on the effect of σ on the total execution time and on the amount of discarded points. The synthetic image in Fig. 2.11 is composed of large homogeneous regions; a low value of σ is enough to achieve a significant decrease in execution time. Due to the high information content of real-world images, the maximum error-free value of σ for the cars image (the one that maximizes the amount of filtered points while not producing any false negative) is higher. Furthermore, the steep rise in saved time and amount of discarded points starts later.

The maximum error-free threshold value for both images in the latter experiment was obtained by means a statistical analysis similar to that of Konishi *et al.* [Konishi et al., 2003]. This analysis involves the definition of two distributions $P(\theta|on)$ and $P(\theta|off)$. The former represents the probability that a pixel is part of the most salient features given its normalized entropy θ at s_{max}. By the other hand, $P(\theta|off)$ represents the probability that a pixel is not part of the most salient features given its normalized entropy θ at s_{max}. In Fig. 3.10 we show these distributions for the two images used in Fig. 3.9. We estimated them from all pixels in each image, **after** applying the Scale Saliency algorithm. The maximum error-free threshold value is given by the first θ value, starting from $\theta = 0$, for which $P(\theta|on) \neq 0$. The fact that Scale Saliency must be applied *a priori* makes this method for estimating the optimal value of σ unfeasible. Furthermore, the maximum error-free threshold may change for different images. Fig. 3.10 also supplies additional evidence of the hypothesis developed in the previous section: pixels with

[2]The total execution time includes the steps 13 and 4 of the filtering algorithm

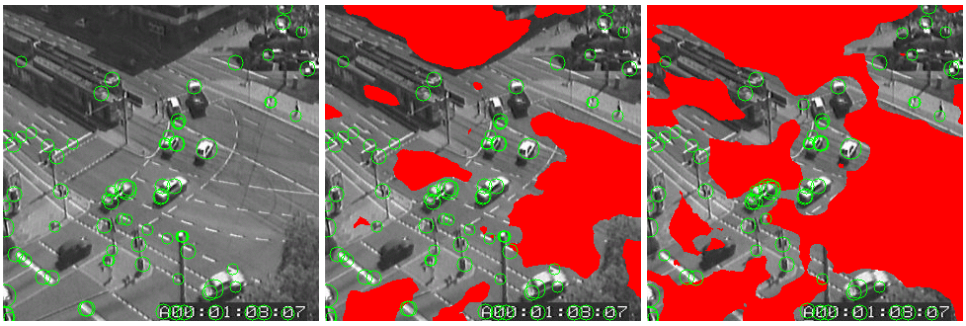


Figure 3.8: Examples of non-salient regions filtering. From left to right: the output of the original Kadir and Brady algorithm, the output of our algorithm using $\sigma = 0.73$ and the output of our algorithm using $\sigma = 0.82$. In all cases, the 200 most salient features (after clustering) were selected. The range of scales was set between $s_{min} = 5$ and $s_{max} = 20$. Filtered points are shown in red.

higher entropy values at s_{max} are more likely to be part of the most salient regions of the image. This evidence is stronger in real-world images.

3.4 Bayesian filtering and Chernoff Information

In this section we introduce the theoretical framework upon which our filtering algorithm is built. The aim is to exploit image statistics in order to estimate a filtering threshold value that is applicable to a set of images. This method relies on the fact that images belonging to a same environment or object category share common properties like intensities, texture patterns, and second order statistics [Torralba and Oliva, 2003][Farinella et al., 2008]. Thus, the entropy values of their most salient features will approximately lays in the same range. Given a set of images belonging to the same category or environment, and following the statistical training approach of Konishi *et al.* [Konishi et al., 2003] (which was later extended by Cazorla *et al.* [Cazorla et al., 2002][Cazorla and Escolano, 2003]), we learn a threshold value for that set. The method is based on estimating the distributions $P(\theta|on)$ and $P(\theta|off)$ for the training set, as we did it in Section 3.3 for individual images (see Fig. 3.10).

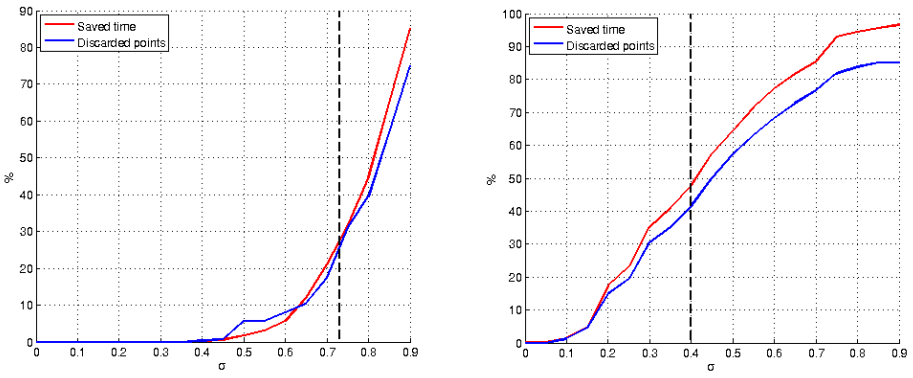


Figure 3.9: Saved time and number of discarded points for the cars image (left) and the synthetic image in Fig. 2.11 (right) as we increase the σ value. The range of scales is set between $s_{min} = 5$ and $s_{max} = 20$. The vertical bold dashed line represents the highest threshold that do not produce false negatives, when the 200 most salient features are selected.

Our method is only feasible in the case of homogeneous sets of images. A question may arise: how can we test if a set of images is homogeneous enough in order to extract useful statistics from it? To that end, we propose to apply the Chernoff Information measure [Cover and Thomas, 1991]. The Chernoff Information $C(P,Q)$ between two probability distributions P and Q is defined as

$$C(P,Q) = -\min_{0\leq\lambda\leq 1} \log\left(\sum_{j=1}^{J} P^{\lambda}(y_j)Q^{1-\lambda}(y_j)\right) \quad , \tag{3.3}$$

where $\{y_j : j = 1,\ldots,J\}$ are the variables over which the distributions are defined. Chernoff information measures the easiness to discriminate between two probability distributions. In Fig. 3.11 we show the $P(\theta|on)$ and $P(\theta|off)$ distributions for two different image categories of the *Visual Geometry Group* dataset, along with their corresponding Chernoff Information value. We estimated these distributions after applying the Kadir and Brady algorithm to all pixels of a set of training images selected from these two categories. Chernoff Information is high when $P(\theta|on)$ and $P(\theta|off)$ are clearly separable, that is, when given the entropy value of a pixel at s_{max}, it is

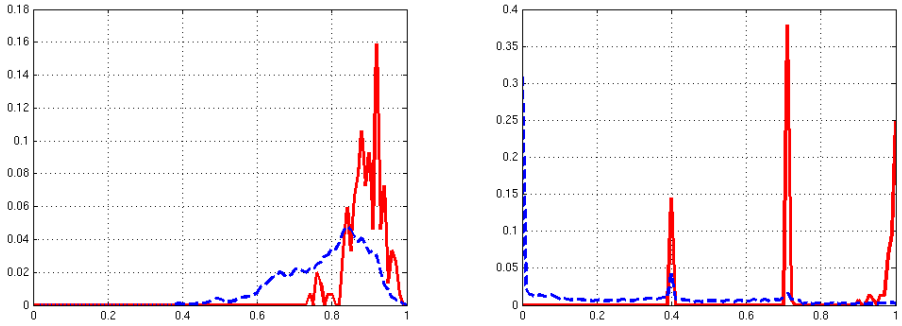


Figure 3.10: Distributions $P(\theta|on)$ (red) and $P(\theta|off)$ (blue) for the cars image (left) and the synthetic image (right).

easy to know if that pixel is part of the most salient features of the image. If Chernoff Information is low, then the image class is not homogeneous enough, and it should be split into homogeneous subsets. In this thesis we do not address the point of selecting an optimal Chernoff Information value for class partition. It is planned as part of our future work (see Section 3.8).

We describe now how to select a filtering threshold value for an image category from its distributions $P(\theta|on)$ and $P(\theta|off)$. This kind of analysis was previously applied to the edge detection and grouping problem [Konishi et al., 2003][Cazorla and Escolano, 2003]. In this problem $P(\theta|on)$ and $P(\theta|off)$ represent the conditional probability that a pixel is in an edge (or not) depending on the response of a particular filter θ. On the basis of the road tracking method proposed by Geman and Jedynak [Geman and Jdeynak, 1996], Konishi *et al.* stated that the log-likelihood ratio of these two distributions can be used as a measure of edge strength, that is, a measure of edgeness [Konishi et al., 2003]. Based on this reasoning, they applied the log-likelihood test

$$\log \frac{P(\theta|on)}{P(\theta|off)} > T \ , \tag{3.4}$$

being T a threshold, in order to determine if a pixel is in an edge given the filter response θ. We propose to utilize this log-likelihood test to discard non-salient pixels before actual application of the Scale Saliency algorithm. In our case θ is the normalized entropy of a pixel at s_{max} with respect to

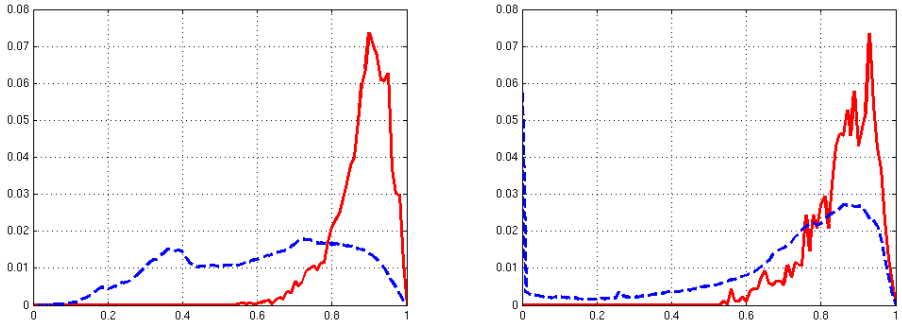


Figure 3.11: Distributions $P(\theta|on)$ (red) and $P(\theta|off)$ (blue) for the *airplanes_side* image category (left) and the *camel* image category (right), from the *Visual Geometry Group* dataset. The Chernoff Information value in the first case is $C(P(\theta|on), P(\theta|off)) = 0.40$ and in the second case is $C(P(\theta|on), P(\theta|off)) = 0.14$. A pixel classifier for the *airplanes_side* category will distinguish better between salient and non-salient pixels.

the maximum entropy at s_{max} in that image. If $T \geq 0$ all pixels for which $P(\theta|on) < P(\theta|off)$ will be filtered. On the contrary, if $T < 0$ we will keep several pixels that are not likely to be part of the most salient features in the image. Once again, a filtering threshold must be chosen. In this sense, it could seem that this method has the same problem as the first filtering algorithm presented in Section 3.3. However, the advantage of Eq. 3.4 is that we can estimate a range of valid T values.

From [Cazorla and Escolano, 2003] we know that the threshold T must satisfy

$$-D(P(\theta|off)||P(\theta|on)) < T < D(P(\theta|on)||P(\theta|off)) \ , \tag{3.5}$$

where $D(p||q)$ is the Kullback-Leibler divergence between distributions p and q:

$$D(p||q) = \sum_{j=1}^{J} p(y_j) \log \left(\frac{p(y_j)}{q(y_j)} \right) \ . \tag{3.6}$$

Kullback-Leibler divergence or relative entropy measures the dissimilarity between two distributions. It can be interpreted as an estimation

of the coding efficiency loss if we assumed that the underlying distribution was q when the actual distribution is p [Cover and Thomas, 1991]. In statistics, it results from the expected logarithm of the likelihood ratio.

Different threshold values in the range given by Eq. 3.5 produce different filtering results. If we take the minimum value in this range, then we get a conservative filter that ensures a good trade-off between a low error rate (low amount of false positives or negatives) and a low computation time. If higher T values are selected, both computational efficiency and error rate increase. The Information Theory measures discussed in this section allow to estimate how the statistics of an image category affect the error rate during filtering. The first important fact is that the overall probability of error increases exponentially with $C(P(\theta|on), P(\theta|off))$ [Cover and Thomas, 1991]. Furthermore, Chernoff Information and Kullback-Leibler measures are related. As stated before, if the value of $C(P(\theta|on), P(\theta|off))$ is low, then the two distributions are similar and it is more difficult to select a threshold value that clearly discriminates between salient and non-salient pixels. Consequently, the range defined in 3.5 is narrower; less points can be filtered and error rate is higher.

An additional advantage of the log-likelihood test is that, in general, its filtering performance is higher. In Fig. 3.12 we compare the filtering results of the log-likelihood test with those of the simple test described in Section 3.3. In the former case, we learned a *general* threshold range for the *bottles* category (see Section 3.5) and we took the highest error-free threshold value in that range (i.e the highest threshold value that did not produce any false negative or positive). More points were filtered than in the latter case, in which we used the highest error-free threshold *for that image*.

3.5 Bayesian filtering of Scale Saliency: the algorithm

In this section we summarize our filtering algorithm. It is based on the concepts introduced in the previous section. Assuming that the input images are divided into classes or categories, the first step involves learning a threshold value for each of these categories. Given a new input image belonging to any of these categories, our algorithm uses the corresponding

Figure 3.12: Left: output of the Scale Saliency algorithm. Center: output of the filtered Scale Saliency algorithm based on the simple test described in Section 3.3, using $\sigma = 0.7$ (23% of the image pixels were filtered). Right: output of the filtered Scale Saliency algorithm based on the log-likelihood test, using $T = -0.65$ (26% of the image pixels were filtered).

threshold value in order to discard pixels that are not likely to be salient before applying the Kadir and Brady method. Our filtering algorithm remarkably decreases computation time with low error rate (see Section 3.6).

The steps of the training phase, in which a threshold value for an image category is learned from a training set, are as follows:

1. Estimate the $P(\theta|on)$ and $P(\theta|off)$ distributions from all the pixels in the set of training images, after testing if these pixels are part (on) or not (off) of the final extracted set of salient features in its corresponding image, and being

$$\theta = \frac{H_D(\mathbf{x}, s_{max})}{\max_{\mathbf{x}}\{H_D(\mathbf{x}, s_{max})\}} \tag{3.7}$$

   the normalized entropy value of a pixel $\mathbf{x}$ at s_{max}.

2. Analyze the value of Chernoff Information between these two probability distributions. If $C(P(\theta|on), P(\theta|off))$ is low, the image class is not homogeneous enough and a valid threshold can not be learned.

In this case, split the image class into several subclasses and repeat the learning phase for each of them.

3. Calculate the Kullback-Leibler divergences $D(P(\theta|off)||P(\theta|on))$ and $D(P(\theta|on)||P(\theta|off))$.

4. Select a threshold in the range $-D(P(\theta|off)||P(\theta|on)) < T < D(P(\theta|on)||P(\theta|off))$. The minimum valid T value provides a conservative trade-off between efficiency and error rate. Higher T values will increase error rate depending on $C(P(\theta|on), P(\theta|off))$ [Konishi et al., 2003].

After the training phase, new input images can be filtered before applying the Scale Saliency algorithm. During the filtering phase our algorithm discards pixels that are not likely to be part of the final set of most salient features in the image:

1. Calculate the local normalized entropy $\theta_{\mathbf{x}}$ at scale s_{max} for each pixel $\mathbf{x}$ using Eq. 3.7.

2. Select the interest points

$$X = \left\{ \mathbf{x} \,|\, \log \frac{P(\theta_{\mathbf{x}}|on)}{P(\theta_{\mathbf{x}}|off)} > T \right\}, \tag{3.8}$$

where T is the learned threshold for the class of the input image.

3. Apply the Scale Saliency algorithm only to $\mathbf{x} \in X$.

We will show examples of application in section 3.6. We will conduct several experiments aimed to demonstrate the validity of our approach. We will also discuss the effect of the algorithm parameters on final results.

3.6 Experimental results

In order evaluate our approach, we conducted a series of experiments using the *Object Category* dataset from the *Visual Geometry Group* (see Section 3.2). The dataset is composed of several image categories (e.g. *airplanes_side* and *faces*), that in turn are composed of a different number of

images, with different sizes. We kept the original categories even though our experiments demonstrated that Chernoff Information was low for some of them, that is, that several categories are not homogeneous enough. In these cases the performance of the algorithm could increase after splitting these homogeneous categories.

In order to learn the range of valid thresholds for each category and to estimate their $P(\theta|on)$ and $P(\theta|off)$ distributions, we built a training set by randomly selecting a 10% of the images in each category (see Section 3.6.1). The range of scales was set between $s_{min} = 5$ and $s_{max} = 20$. Input images were scaled before the application of the algorithm; as a result, all images have a maximum height or width of 320 pixels. Regarding entropy estimation, we used 128-bin histograms instead of 256-bin histograms. The histogram quantization yields a remarkably lower execution time and it does not affect the final results to a great extent. Kadir and Brady also use histogram quantization in their implementation of the Scale Saliency algorithm[3] (they use 64-bin histograms). The amount of extracted salient features was set to 50. However, the final number of features is in general lower, due to the non-maximum suppression step (see Section 2.2.2). We set the features clustering parameters to $K = 3$ and $V_{th} = 70$.

Table 3.2 shows the results for the aforementioned dataset. We applied two different thresholds to each image category: the minimum T value in the range $-D(P_{off}||P_{on}) < T < D(P_{on}||P_{off})$ and $T = 0$. The *% points* column shows the mean amount of points discarded for each image before Scale Saliency feature extraction, and the *% time* column indicates the mean saved time when comparing the filtered Scale Saliency algorithm (that includes both the filtering step and the Scale Saliency application to the rest of pixels in the image) to the non-filtered method. Finally, the last column shows the mean error rate for each image category. This error is calculated as

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}\frac{d(A_i, B_i) + d(B_i, A_i)}{2} \quad , \tag{3.9}$$

where n is the number of images in that category, $A = \{A_1, \ldots, A_n\}$ is the set of the most salient regions obtained by means of the Scale Saliency algorithm for each image in the category (after non-maximum suppression), $B =$

[3] http://www.robots.ox.ac.uk/~timork/salscale.html

$\{B_1, \ldots, B_n\}$ is the set of the most salient regions obtained by means of our approach for each image in the category (after non-maximum suppression) and

$$d(A, B) = \sum_{a \in A} \min_{b \in B} ||a - b|| \ , \tag{3.10}$$

is a metric based on the Euclidean distance that measures the distance between the feature sets A and B. This distance is calculated in $\mathcal{R}^3$ (location and scale): the distance in the scale-space is also considered.

Better results were obtained for those categories for which Chernoff Information value was higher: generally more points are discarded or mean error is lower. For instance, the highest Chernoff Information values were attained for the *airplanes_side*, *cars_brad_bg* and *leaves* categories. In these cases, using the minimum valid T for each of them saves up to a $35 - -40\%$ of time. The lowest Chernoff Information values correspond to the *bottles*, *camel* and *google_things* categories (the latter one, in particular, is the most heterogeneous of all; its function is to model clutter). However, even for these categories the mean saved time is high ($20 - -25\%$) and we rarely found any image for which the time of the complete process was higher than that of the Scale Saliency algorithm. On the other hand, the threshold value $T = 0$ yields remarkable improved results for all categories while maintaining a low error rate. The value of T will depend on the application. The minimum valid value in the range defined by Eq. 3.5 is a good trade-off between speed and low error rate. The value $T = 0$ could be used in tasks like robot localization, in which computational efficiency is a key factor [Newman et al., 2006].

In Fig. 3.13 we show several examples of application of our filtering algorithm. As one may expect, uniform background is almost completely filtered. Nevertheless, textured and non-uniform regions can also be discarded due to the fact that our algorithm is based on normalized entropy and that a different threshold is learned for each image category. Furthermore, the mean amount of discarded points is not necessarily higher for those image categories containing images with uniform background. For instance, almost all images in the *bottles* category have an uniform background, but results for this category are worse than those of the *houses* category, in which the presence of textured regions is higher.

Test set	Chernoff	T	% points	% time	ϵ
airplanes_side	0.415	-4.98	30.79%	42.12%	0.0943
		0	60,11%	72.61%	2.9271
background	0.208	-2.33	15.89%	24.00%	0.6438
		0	43.91%	54.39%	5.0290
bottles	0.184	-2.80	9.50%	20.50%	0.4447
		0	23.56%	35.47%	1.9482
camel	0.138	-2.06	10.06%	20.94%	0.2556
		0	40.10%	52.43%	4.2110
cars_brad	0.236	-2.63	24.84%	36.57%	0.4293
		0	48.26%	61.14%	3.4547
cars_brad_bg	0.327	-3.24	22.90%	34.06%	0.2091
		0	57.18%	70.02%	4.1999
faces	0.278	-3.37	25.31%	37.21%	0.9057
		0	54.76%	67.92%	8.3791
google_things	0.160	-2.15	14.58%	25.48%	0.7444
		0	40.49%	52.81%	5.7128
guitars	0.252	-3.11	15.34%	26.35%	0.2339
		0	37.94%	50.11%	2.3745
houses	0.218	-2.62	16.09%	27.16%	0.2511
		0	44.51%	56.88%	3.4209
leaves	0.470	-6.08	29.43%	41.44%	0.8699
		0	46.60%	59.28%	3.0674
motorbikes_side	0.181	-2.34	15.63%	27.64%	0.2947
		0	38.62%	51.64%	3.7305

Table 3.2: Results for the *Object category* dataset

Figure 3.13: Examples of filtering for three images belonging to three different image categories (from top to bottom: *houses*, *google_things* and *motorbikes_side*). Red regions represent discarded points. From left to right: results of the Scale Saliency algorithm, results using the minimum valid T value and results using $T = 0$.

We also applied our algorithm to the *Caltech101 database*[4]. This database consists of 101 image categories. Each category contains from 40 to 800 images (see Fig. 3.14). We randomly selected 20 training images and 20 test images per category. The training images were used to learn a range of valid threshold values for each category. Finally, we computed the mean amount of filtered points and the mean average error rate for each category using two threshold values: the minimum valid one for that category, and $T = 0$. The experiment was repeated 10 times. The results can be seen in Fig. 3.15. These results are also summarized in Table 3.3, where we show the categories for which the algorithm yields the maximum and minimum error rate and amount of filtered points.

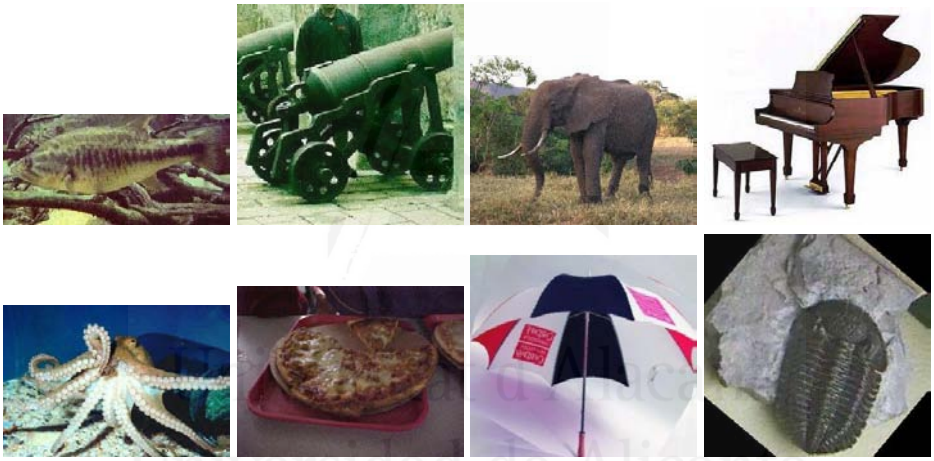


Figure 3.14: Example images from the *Caltech101* dataset.

In general the mean saved time for a category is negative if the amount of filtered points is below 20% for that category. In these cases the execution time of complete process, including the filtering step, is higher than that of the original Scale Saliency algorithm. However, this is the case of less than the 40% of the *Caltech101* dataset categories, and only for $T = min$. If $T = 0$ our approach always decreases execution time. Regarding error, those categories for which more pixels are filtered do not necessarily correspond to those categories for which the error rate is higher. There are exceptions, like for instance the *stapler* category in the $T = 0$ experiment, but it must be

[4]http://www.vision.caltech.edu/Image_Datasets/Caltech101/

noted that up to an average of 82.10% filtered points is achieved. As one may expect, lower error rates do not correspond to highest amounts of filtered pixels, except in several categories like the *Face* category (which error rate is 0 and is the seventh category in order of most filtered points for $T = min$).

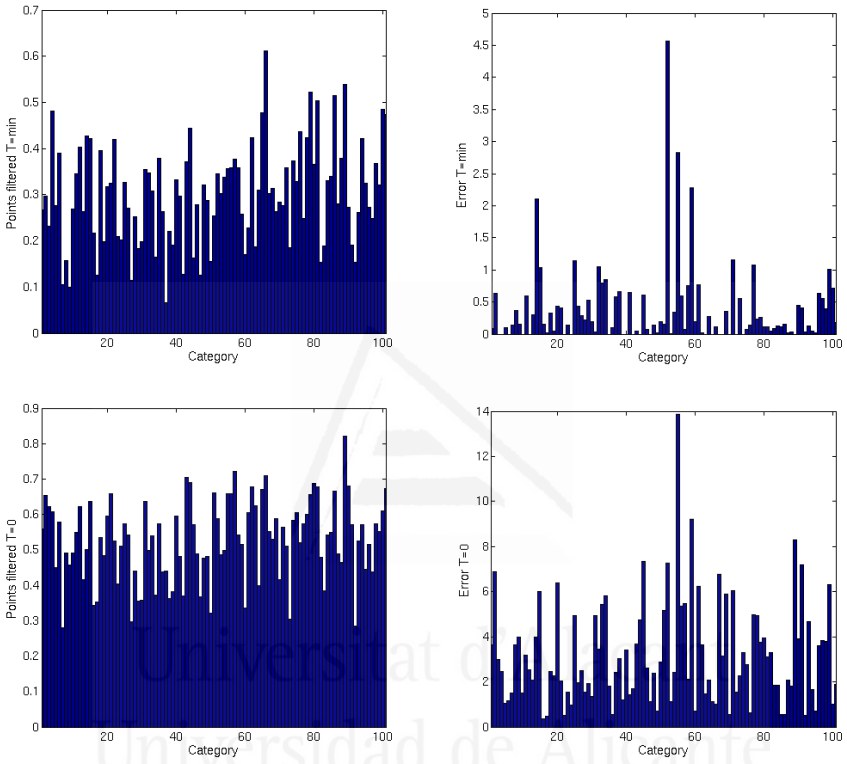


Figure 3.15: Experimental results using the *Caltech101* dataset. In all plots the results of the 101 categories are represented along the x direction. Top: average amount of filtered points and average error rate per category, being T the minimum valid threshold for each category. Bottom: average amount of filtered points and average error rate per category, being $T = 0$.

3.6.1 Training limits

In the experiments with the *Visual Geometry Group* dataset summarized in the previous section the 10% of the images in each category were randomly selected for training, that is, to learn their corresponding category thresholds.

	$T = min$	$T = 0$
Most filtered points	mayfly(66): 61.23% stapler(89): 53.85% revolver(79): 52.12% seahorse(86): 51.51% rooster(81): 50.34%	stapler(89): 82.10% joshua_tree(57): 72.13% mayfly(66): 71.04% ewer(43): 70.54% ferry(44): 69.02%
Less error	cannon(22): 0 Faces(4): 0 Leopards(6): 0 anchor(10): 0 barrel(12): 0	bonsai(16): 0.34 brain(17): 0.46 cannon(22): 0.51 stop_sign(92): 0.52 scorpion(85): 0.54
Less filtered points	dolphin(37): 6.69% airplanes(9): 9.93% motorbikes(7): 10.43% chandelier(27): 11.53% brain(17): 12.57%	motorbikes(7): 28.12% stop_sign(92): 28.43% chandelier(27): 29.74% pagoda(73): 30.42% grand_piano(50): 32.20%
Most error	headphone(52): 4.55 ibis(55): 2.82 ketch(59): 2.27 beaver(14): 2.10 octopus(71): 1.16	ibis(55): 13.85 ketch(59): 9.22 stapler(89): 8.30 flamingo(45): 7.32 headphone(52): 7.27

Table 3.3: Summary of the results shown in Fig. 3.15, for the minimum valid threshold of each category ($T = min$) and for $T = 0$. The index of the categories is shown in brackets. Several categories had zero error for $T = min$, so only a subset of them is shown in this table.

We chose that proportion of training images after conducting an experiment aimed to explore the limits of the training step, that is, to estimate the range of values in which we may set this parameter. The experiment involved the analysis of the evolution of $C(P(\theta|on), P(\theta|off))$ for each image category, depending on the amount of training images. Chernoff Information has been previously used to measure the quality of a classifier [Konishi et al., 2003]. Furthermore, in our case (as we stated in Section 3.5) this measure is related to the quality of the filtering process. These facts motivated the use of Chernoff Information in this experiment.

We repeated the training step for all the image categories in the *Visual Geometry Group* dataset using different proportions of training images. Fig. 3.16 represents the evolution of Chernoff Information for all categories, when selecting from 2% to 30% of images per category. In almost all cases, Chernoff Information decreases or changes sharply in the range between 2% and 10%. From 10% it varies smoothly or remains constant. Our interpretation of this plot is that at least a $10 - 15\%$ of training images per category is required. If the proportion is lower, there is not enough information to extract a correct threshold: Chernoff Information is too high (the range defined by Eq. 3.5 is too wide, meaning that we can select thresholds that discard too many pixels or almost none), or it changes randomly. A proportion higher than 15% does neither decrease filtering performance nor provide additional information.

3.6.2 The effect of the number of most salient features

The width of the range depends on the Kullback-Leibler divergence between $P(\theta|on)$ and $P(\theta|off)$, that in turn depends on the amount of most salient features extracted by the algorithm. Indeed, if more salient features are to be extracted, less points can be filtered: as this parameter increases, the range of normalized entropies of the most salient features in the image grows, that is, the minimum θ value for which $P(\theta|on) > 0$ is lower (see Fig. 3.17). Therefore, the overlap area between $P(\theta|on)$ and $P(\theta|off)$ increases: the Chernoff Information between these distributions decreases and also the range of valid T values gets narrower.

We conducted a series of experiments on the *Visual Geometry Group*

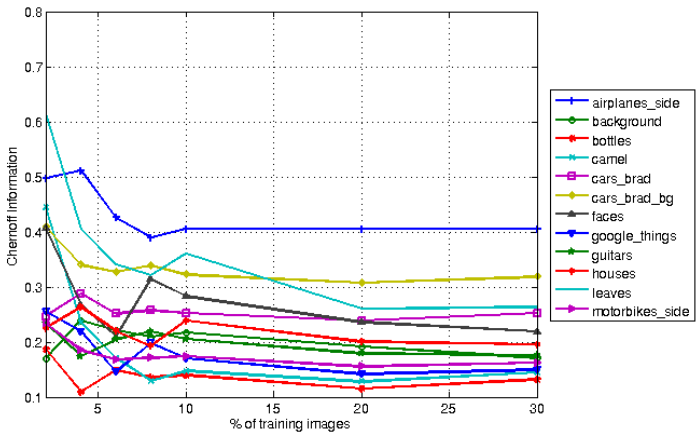


Figure 3.16: Evolution of $C(P(\theta|on), P(\theta|off))$ for all image classes in the *Visual Geometry Group* dataset, depending on the % of images used for training.

dataset to demonstrate this fact. We show the results for the *bottles* category in Table 3.4 and Fig. 3.18. The effect of changing the amount of salient features to be extracted is not as noticeable as in the case of one image (Fig. 3.17). As we increase the number of extracted features, the value of Chernoff Information slightly decreases from 0.184 to 0.172. Then, it increases again from 0.171 to 0.176. These results confirm our hypothesis: if the amount of extracted salient features is low, it is easier to discriminate between interesting and non-interesting points (that is, points that can be filtered) due to the fact that Chernoff Information is higher. Thus, more points can be filtered. On the other hand, we observed that for a high number of extracted features, Chernoff Information also increases: more pixels with high normalized entropy at s_{max} become part of the $P(\theta|on)$ distribution and the overlap region between $P(\theta|on)$ and $P(\theta|off)$ is smaller.

However, in spite of the results shown in Table 3.4, this parameter slightly affects the final results. An example is shown in Fig. 3.19. As can be seen, using the minimum valid T value, the proportion of filtered points remains practically constant for a different amount of extracted salient features.

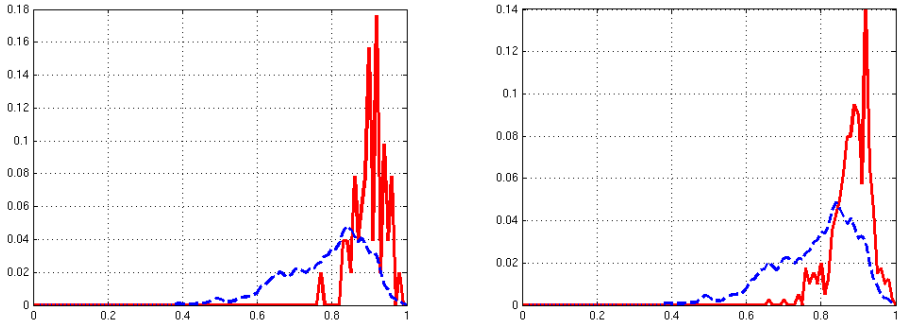


Figure 3.17: Distributions $P(\theta|on)$ and $P(\theta|off)$ for the cars image, if the 50 most salient features are extracted (left, $C(P(\theta|on), P(\theta|off)) = 0.568682$) and if the 400 most salient features are extracted (right, $C(P(\theta|on), P(\theta|off)) = 0.290655$).

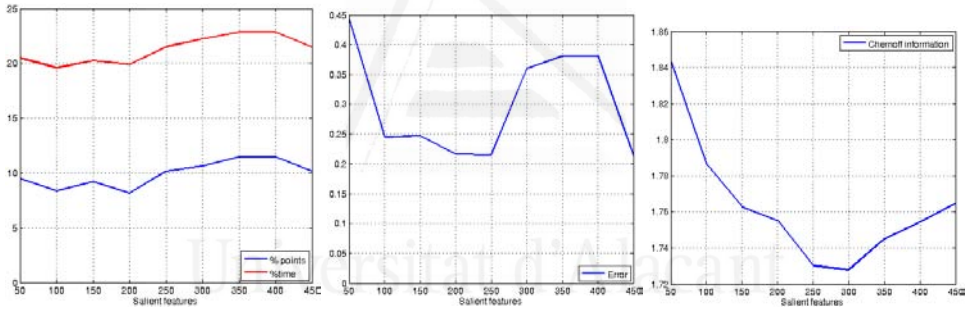


Figure 3.18: Experimental results for the *bottles* image category, when extracting an increasing number of most salient features.

3.6.3 The effect of the range of scales

The range of scales is a key parameter in the Scale Saliency algorithm. It directly affects execution time. Thus, this parameter should be set after analyzing the range of scales in which the most salient features of a set of images are located. For instance, in Fig. 3.20 we show that the most salient features of the *Object category* dataset for $s_{min} = 5$ and $s_{max} = 20$ are not scattered in the scale-space; on the contrary, the scales of the most salient features usually fall in the range $[5, 15]$. We decided to repeat the experiments for the *Object category* dataset using $s_{min} = 5$ and $s_{max} = 20$.

Displayed	Chernoff	T	% Points	% Time	ϵ
50	0.184404	-2.80	9.50%	20.50%	0.4447
100	0.178672	-2.69	8.38%	19.58%	0.2448
150	0.176282	-2.64	9.18%	20.24%	0.2468
200	0.175527	-2.61	8.16%	19.91%	0.2159
250	0.173012	-2.54	10.16%	21.51%	0.2151
300	0.172778	-2.52	10.62%	22.25%	0.3601
350	0.174482	-2.53	11.45%	22.87%	0.3811
400	0.175427	-2.53	11.45%	22.87%	0.3811
450	0.176483	-2.54	10.16%	21.51%	0.2151

Table 3.4: Experimental results for the *bottles* image category, using different number of most salient features extracted.

In Table 3.5 we show the results of our filtered Scale Saliency algorithm for this narrower range of scales. In all cases, the mean amount of discarded points increased from two to five times when compared to those results in Table 3.2 (and, as a consequence, the mean saved time also increased). At least a 55% of time is saved for any image class. Error rate did not necessarily increase; in fact, it was lower for several image categories. This experiment demonstrates that a previous scale-range analysis, specific to the used dataset, can remarkably improve the efficiency of Scale Saliency.

It must be noted that the result of the scale-space analysis should be a *range of scales*, and not a sparse set of scales. By estimating saliency through a range of scales we can significantly increase computational efficiency by means of *cumulative histograms*[5]. Recall that in order to estimate the saliency of a pixel at a given scale we build an intensity frequency histogram from that pixel neighbourhood, being the scale the radius of this circular neighbourhood. As we increase the scale, the histogram includes new pixels, but also those pixels that were also included at previous scales. The cumulative histograms technique is based on storing the unnormalized intensity histogram for all the scales. Therefore, the normalized histogram

[5]as can be seen in the code provided by Kadir and Brady, available at http://www.robots.ox.ac.uk/∼timork/salscale.html

Figure 3.19: Filtering results for the same image (taken from the *bottles* image category), using the minimum valid T learned in the case of different number of extracted salient features (from left to right and from top to bottom: 50, 150, 250 and 350 extracted features). Red regions represent discarded points.

for scale s is computed after adding the intensity of the new pixels to the unnormalized histogram for scale $s-1$, and then normalizing. As can be seen, if we explore a sparse set of scales, instead of a complete a range of scales, cumulative histograms can not be applied. In this case the execution time is strongly affected and the Scale Saliency algorithm is highly inefficient.

3.7 Application: robot localization

The most noticeable limitation of our filtering algorithm is that it requires *a priori* knowledge of the category or environment to which an image belongs. This limitation may prevent us from applying the algorithm to real-world problems, like image categorization. In spite of this fact, our algorithm can be useful in other contexts. In this section we show how we successfully applied the filtered Scale Saliency method to the robot localization problem. The results shown in this section were obtained in collaboration with researchers in this field [Escolano et al., 2007].

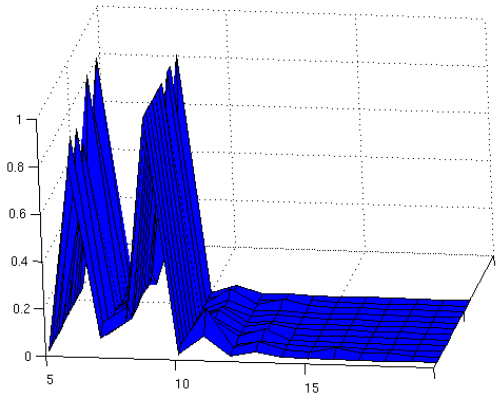


Figure 3.20: Frequency of most salient regions between $s_{min} = 5$ and $s_{max} = 20$ for all image the image categories in the *Object category* dataset. If we applied Scale Saliency in the range of scales between $s_{min} = 5$ and $s_{max} = 15$, a low number of actual salient features would be discarded.

The problem of robot localization (see [Arkin and Balch, 1998] and [Thrun et al., 2001]) is summarized as follows: A camera mounted on a robot is used to capture images from an environment in order to build an image database. Then, the robot is placed somewhere in that environment. During robot operation, and in order to find out its approximate location, the robot captures a new image and searches for the most similar one in the database. This image is the output of the algorithm. Robot localization has been considered one of the most important capabilities of fully autonomous robots [Cox, 1989]. The environment of our robot localization experiment is shown in Fig. 3.21. This map represents the 3D reconstruction of some areas of the Politécnica III building in the University of Alicante and its surroundings. It was obtained by means of a 6-DOF SLAM (Simultaneous Localization And Mapping) algorithm [Sáez and Escolano, 2006].

Our robot localization method relies on a coarse-to-fine search. The global environment is manually divided into six submaps or categories, corresponding to six different environments (see Fig. 3.22). The system stores a database of 721 images, covering the whole global environment. The images in the database identify possible poses of the robot in the environment. The

Test set	Chernoff	T	% Points	% Time	ϵ
airplanes_side	0.508	-5.69	68.05%	75.63%	2.3041
background	0.288	-3.27	48.87%	54.33%	3.0359
bottles	0.252	-3.61	47.66%	55.07%	3.7397
camel	0.193	-2.75	51.16%	58.58%	3.8124
cars_brad	0.320	-3.50	62.03%	70.12%	3.9246
cars_brad_bg	0.454	-4.97	69.10%	77.08%	4.6106
faces	0.369	-4.22	66.07%	74.00%	7.8287
google_things	0.220	-2.71	52.00%	59.37%	4.9537
guitars	0.315	-3.63	55.37%	62.82%	2.8483
houses	0.312	-3.77	55.98%	63.34%	3.1287
leaves	0.603	-7.11	63.46%	71.24%	3.2143
motorbikes_side	0.267	-3.45	48.98%	56.99%	2.9920

Table 3.5: Results for the *Object category* dataset with a narrower range of scales

coarse step estimates the submap where the robot is located. In this step we perform a k-nearest neighbor classification based on simple image filters that are applied to the input image. Its output is the set of k database images that are most similar to it [Bonev et al., 2007a].

The **fine step** is based on feature extraction. The system extracts salient features from the input image by means of our filtered Scale Saliency approach. A SIFT descriptor is computed for each feature. A SIFT descriptor [Lowe, 2004] is a 128-dimensional feature vector that represents the appearance of a salient region in an image. This representation is invariant to scale and rotation. SIFT and SIFT-based descriptors are commonly used in Computer Vision applications due to their high matching performance [Mikolajczyk and Schmid, 2004a]. We match the features extracted from the original image with those extracted from the k selected database images. The matching step relies on structural criteria in order to remove inconsistent matches [Aguilar, 2006]. Finally, the output of the localization algorithm is the database image, from the k selected images during the coarse step, that produces the highest amount of matches with the

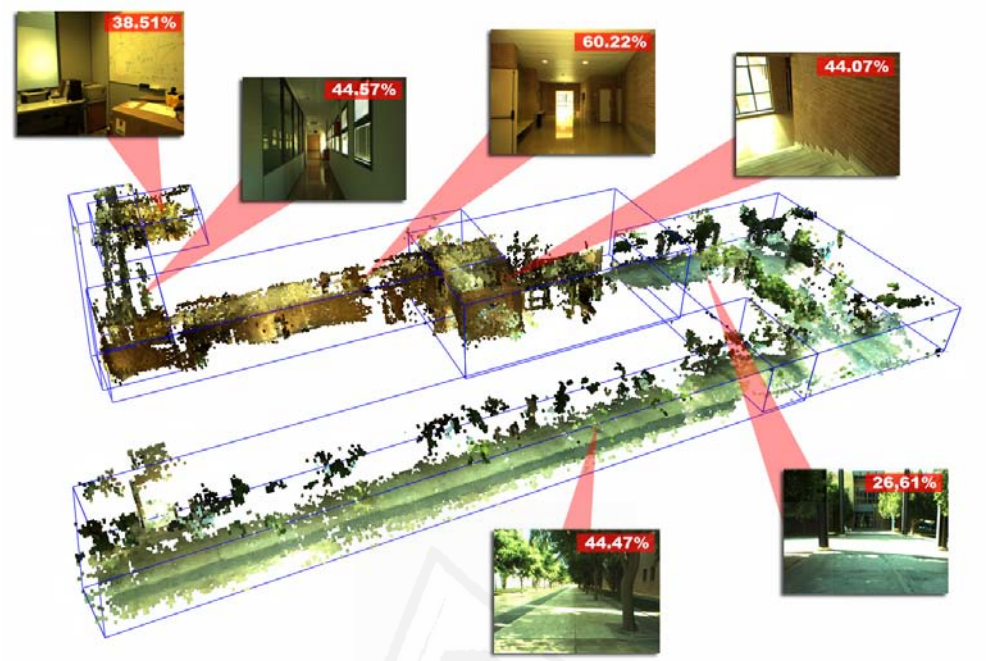


Figure 3.21: 3D reconstruction of some parts of the Politécnica III building in the University of Alicante, obtained with 6-DOF SLAM. We show an example image to represent each category along with its mean proportion of discarded points.

input image.

The purpose of the coarse step is twofold. Firstly, it selects a small set of image candidates; thus, the fine step is not applied to the whole database. And secondly, it detects the submap where the robot is located. This way, we know the category of the input image during the fine step and we can use a previously learned threshold to filter image points before matching. In Table 3.6 we show the results of the experiment, and in Fig. 3.21 we summarize these results. In Table 3.6 we can see, for each submap, the Chernoff Information value, the minimum valid threshold value (that is the value that we used during filtering) and the mean proportion of filtered points. This proportion spans from 26% to 60%, depending on the submap. Several examples of filtered Scale Saliency application are shown in Fig. 3.23.

Figure 3.22: Example images from the 6 environments in the robot localization experiment. From left to right and from top to bottom: *office*, *corridor1*, *corridor2*, *hall*, *entrance* and *trees-avenue*.

3.8 Conclusions

In this chapter we have introduced a filtering approach aimed to discard non-interesting points before applying the Scale Saliency algorithm. It is based on setting a normalized entropy threshold at the maximum scale. This threshold is inferred from the analysis of a set of images belonging to the same environment or category. We tested the effect of the parameters of the algorithm and showed its performance. We also showed a practical application in the robot localization field. The main contributions of this chapter are:

- We studied the evolution of Shannon's entropy in the image and scale-spaces. We demonstrated that the entropy of a pixel at the maximum scale puts a bound on the maximum saliency of that pixel in scale-space. This fact supports the hypothesis that homogeneous or non-salient points at higher scales are probably non-salient at lower scales.

- We proposed a simple filtering approach. Unfortunately, it is not feasible: in order to avoid false positives or negatives, the filtering threshold can only be set *after* applying the Scale Saliency algorithm.

Environment	Chernoff	T	% Points
office	0.8977	-9.4877	38.51%
corridor1	0.2482	-2.8053	44.57%
corridor2	0.6518	-7.4953	60.22%
hall	0.5694	-7.4915	44.07%
entrance	0.2859	-3.9325	26.61%
trees-avenue	0.8543	-8.6893	44.47%

Table 3.6: Experimental results of the robot localization experiment.

However, the algorithm demonstrates that it is possible to filter a high amount of pixels using only information from the maximum scale. Furthermore, it is the basis of our final filtering algorithm.

- We showed that a set of images belonging to a same environment or category can be grouped together in order to learn a filtering threshold that is valid for new and unseen images belonging to that category.

- We exploited Information Theory in order to evaluate the applicability of our approach to an image category (Chernoff Information) and to set a range of valid threshold values for that category (Kullback-Leibler divergence). Both are related and they give an approximate measure of error rate.

- The requirement of knowing the category of an image prior to the application of the filtered Scale Saliency could seem a strong limitation of our algorithm. However, we showed a practical application of it in the robot localization field. Other possible application could be feature extraction and tracking in video surveillance images. In this case the environment to which the images belong is also known.

We see two main ways to improve the approach presented in this chapter. The first one is related to the partition of images into categories. During our experiments we kept the categories of the *Object category* and *Caltech101* datasets, and we manually divided the images into categories in the the robot localization experiment. In the latter case we tried to

cluster training images by means of simple filters, similar to the ones used during the coarse step of the robot localization algorithm. The images were represented as vectors (built from filters output) and clustered using several well-known algorithms, like k-means. Although the Chernoff Information of unsupervised categories was higher (in fact, only 5 categories were needed to improve the overall value of Chernoff Information), filtering results for test images were disappointing: the mean proportion of filtered points and the mean saved time did not improve when compared to the manually created categories.

On the other hand, our work may be the first step in discovering new properties of the entropy function in the scale-space. From these results we could devise new filters. Our working hypothesis is that these filters should be organized in cascade, so that each filter processes the output of the previous one (this idea is similar to that of the text recognition algorithm proposed by Chen and Yuille [Chen and Yuille, 2004], that in turn is based on the AdaBoost face classifier proposed by Viola and Jones [Viola and Jones, 2001]). The self-dissimilarity term W_D in particular could be analyzed separately. This term acts as a normalization term that weights entropy values depending on the variation of the local pdf between scales (in our case, the intensities histogram). Therefore, pixels could be discarded regarding the intensity distribution of their neighbourhood.

Figure 3.23: Filtering prior to Scale Saliency applied to images from four different submaps (from top to bottom) in the robot localization experiment. From left to right: output of the Kadir and Brady algorithm, filtered Scale Saliency for the minimum valid T value of each category, filtered Scale Saliency for $T = 0$.

Chapter 4

Multi-dimensional Scale Saliency

4.1 Introduction

In the Scale Saliency algorithm proposed by Kadir and Brady [Kadir and Brady, 2001] entropy and dissimilarity between scales are estimated by means of histograms (see Eq. 2.76). Histogram-based estimation is subjected to the curse of dimensionality: temporal and spatial complexity increases exponentially with respect to data dimensionality. Thus, the Kadir and Brady method is slow when applied to color images, and it is unusable in the case of even higher dimensionalities. Furthermore, as the dimensions of the histogram increase, these histograms are sparser; as a consequence, they are are less informative.

In this chapter we propose a multi-dimensional Scale Saliency approach that can cope with high-dimensional data. We analyze alternative information-theoretic estimation methods which complexity order is linear with respect to data dimensionality and we assess their suitability for the algorithm in terms of computational efficiency and the quality and amount of extracted features.

Firstly, in Section 4.2 we survey several entropy and divergence methods based on Minimal Spanning Trees and K-Nearest Neighbor Graphs. Next, in Section 4.3 we summarize k-d partition, en entropy estimator that recursively partitions the data space following the k-d tree method. We also propose a

new divergence measure based on the k-d partition algorithm and on the total variation distance. In Section 4.4 we present our multi-dimensional Scale Saliency algorithm. It relies on the k-d partition entropy and divergence estimators, due to the conclusions drawn from our experimental results in Section 4.5. These experiments were aimed to test the computational complexity and the quality of estimation of the previously introduced estimators, and also to test the quality and the number of the extracted salient features when using these estimators. Finally, in Section 4.6 we show an application of our multi-dimensional Scale Saliency algorithm to the texture categorization problem.

4.2 Entropy and divergence estimation from entropic graphs

A good overview of non-parametric entropy estimators is given by Beirlant *et al.* [Beirlant et al., 2001]. They divided these techniques into two groups: plug-in estimates and spacing (for 1d) or nearest neighbor (for general dimensionality) estimates. The former involves the estimation of the underlying distribution of the samples in order to plug it into the entropy equation. Entropy estimation from histograms is a clear example of plug-in estimator. Another plug-in approaches are based on kernel density estimation, like Parzen windows [Erdogmus et al., 2004]. The latter group of estimators rely on the calculation of the distance between samples. This is the case of the entropy estimation method proposed by Leonenko *et al.* [Leonenko et al., 2008] and the Rényi α-entropy estimation from Minimal Spanning Trees proposed by Hero *et al.* [Hero and Michel, 1999] (both approaches are explained in this chapter).

The asymptotically convergence of graph-based methods is faster, especially in the case of non-smooth and high-dimensional data [Hero et al., 2003]. Kernel based methods (like Parzen windows estimation) perform worse on or non-smooth densities and are more sensitive to outliers. Furthermore, kernel and histograms methods are subjected to the curse of dimensionality: higher dimensions require a remarkable increase of the amount of samples. In this section we adopt

the term *entropic graph* to refer to the family of minimal graphs spanning a set of samples that produce consistent estimations of entropy and divergence [Hero et al., 2002]. Specifically, we applied entropy estimators based on Minimal Spanning Trees (MSTs) and K-Nearest Neighbor Graphs (KNNGs) [Costa and Hero, 2004].

Our multidimensional extension of the Scale Saliency algorithm is applied to images in which any pixel $\mathbf{x} \in X$ is represented by a d-dimensional vector (3D vectors in the case of RGB images, for instance, or 31D vectors in the case of hyperspectral images with 31 bands). In our implementation the neighborhood $R_{\mathbf{x}}$ of a pixel is represented by an undirected and fully-connected graph $G = (V, E)$, being the nodes $\mathbf{v}_i \in V$ the d-dimensional vectors corresponding to the pixels $\mathbf{x}_i \in R_{\mathbf{x}}$ and E the set of edges connecting each pair of nodes. An edge $e_i \in E$ is labeled with the Euclidean distance in $\mathcal{R}^d$ between the two nodes incident to that edge. The Minimal Spanning Tree (MST) of G is the tree (a subgraph of G that connects any pair of nodes (v_i, v_j) by exactly one path) with minimum total distance with respect to the rest of possible trees spanning V. If $|V| = n$, then $|E| = n-1$. The k-nearest neighbor graph (KNNG) is the subgraph of G that connects each node to its k-nearest neighbors. In this case, $|E| = Kn$. We show in Fig. 4.1 an example of each type of graph.

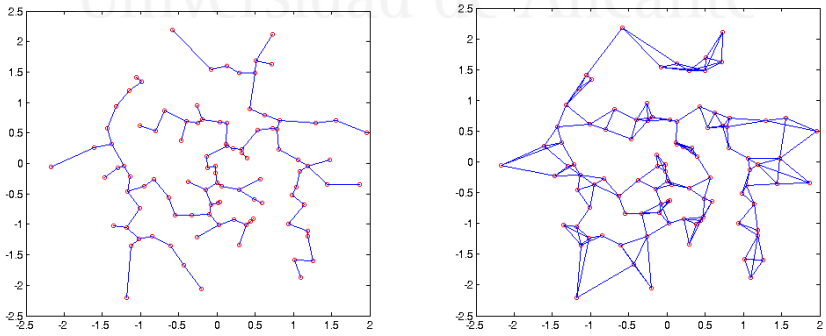


Figure 4.1: Examples of MST (left) and KNNG with $k = 3$ (right) graphs built from the same set of samples.

4.2.1 Entropic graph algorithms

We can find a number of algorithms to build MSTs and KNNGs in the literature. In fact, the first MST algorithms date from the twenties and thirties (although it is now outdated, the survey by Eisner [Eisner, 1997] is a good starting point to study this class of algorithms). In this section we summarize the MST and KNNG approaches used during the experiments in Section 4.5.1.

Firstly we describe two classical MST methods: Kruskal [Kruskal, 1956] and Prim [Prim, 1957] algorithms. In the former one, edges are ordered by weight (by Euclidean distance in our case). Then, edges are taken in order to decide if they must be included in the MST or not. If the result of adding the edge to the MST is the creation of a cycle, it is discarded. Otherwise, it is kept. The latter algorithm also starts with no edges in the MST, but instead of maintaining a set of trees that eventually are merged to build the MST, it grows a single tree. The algorithm starts by selecting the lightest edge leaving an arbitrary initial node. Then, at each step, the lightest edge that leaves the tree is added to it. In both cases we used the *pairing heap* data structure [Fredman et al., 1986] to order edges by weight. Its implementation is simple and, although its asymptotic temporal complexity has proven to be difficult to estimate, experimental comparisons demonstrate that in practice this algorithm is as fast or even faster than other efficient heap algorithms [Stasko, 1987][Moret and Shapiro, 1991].

The Borůvka algorithm [Borůvka, 1926] is a well-known example of bottom-up method. It is easily parallelizable. Firstly, the lightest edge leaving each vertex is selected and added to the MST. Each connected set of nodes forms a subtree. Then, in the following iterations, the lightest edge leaving each subtree is added to the MST and these steps are repeated until the final tree is built. The algorithm includes a contraction step, in which the edges that do not leave from a subtree terminal point are removed. In practice these three algorithms (Borůvka, Prim and Kruskal) are the most used. There are more theoretically efficient algorithms, whose complexities are close to linear (see [Fredman and Tarjan, 1987], [Gabow et al., 1986] and [Pettie and Ramachandran, 2002]). However, their implementation is complex and they are considered impractical due to the effect of the constant factors on their execution time [Katriel et al., 2003].

The computational cost of the algorithm proposed by Katriel *et al* [Katriel et al., 2003] is similar to that of the one proposed by Fredman and Tarjan [Fredman and Tarjan, 1987], but its implementation is simpler and in practice it runs faster. Firstly, a subgraph G' is built from G taking a random set of edges. Then, the algorithm computes the Minimal Spanning Forest (MSF) from G' (G' may not be totally connected), that we call T'. The MSF is the minimal weight set of trees that spans all the nodes of each connected component in the graph. In the next step the algorithm removes the edges $e \in E$ in G that are the maximum weight edge in any cycle in the graph $T' \cup \{e\}$. Finally, the algorithm calculates the MST of G using only the non-removed edges. Both the MSF and MST are built by means of the pairing heap based Prim algorithm. This is the only randomized algorithm that we used during our experiments.

Regarding the KNNG building algorithm, it basically involves finding the k-nearest neighbors of each graph node. In order to search for nearest neighbors we apply the semidynamic version of the k-d tree algorithm proposed by Bentley and Friedman [Bentley, 1990]. The k-d tree data structure [Bentley, 1975][Friedman et al., 1975] is computationally efficient. However, its main drawback is that it is not suitable for high-dimensional data. The number of nodes n in the tree should be $n >> d$; otherwise, most of these nodes are evaluated during nearest neighbor search and the the computational efficiency is similar to the efficiency of exhaustive search [Goodman and O'Rourke, 2004]. It may seem contradictory, as we intend to avoid the curse of dimensionality; nevertheless, the semidynamic k-d tree algorithm allows the update of the tree while exploring the scale-space. Due to this fact we can achieve a decrease of execution time comparable to that of the cumulative histograms in the Kadir and Brady implementation (see Section 3.6.3). As far as we know there is not any dynamic MST algorithm that we can apply to our problem. The most efficient MST updating algorithms that we are aware of require planar graphs [Eppstein et al., 1996] or nodes of degree two or one [Frederickson, 1983].

The semidynamic k-d tree data structure proposed by Bentley [Bentley, 1990] is capable of deleting and undeleting points, but it can not be used to insert new points in the tree. The expected computational time

of the delete, undelete and nearest neighbor search operations is constant. The data structure is also less prone to degenerate cases. The semidynamic k-d tree algorithm works as follows. A new boolean *empty* field is added to the tree nodes. The value of this field is *true* if the subspace represented by the node does not contain any sample. If an empty node is reached during nearest neighbor search, it is immediately abandoned. In order to delete a point from the tree, it is removed from the corresponding leaf node and then the algorithm travels through the tree from bottom to top activating the *empty* field if it is necessary. The experiments of Bentley demonstrate that this method does not require to rebalance the tree due to the fact that the deleted points do not significantly affect the temporal cost of search operations. The undelete operation is similar. The point is assigned again to the corresponding leaf node, and then the algorithm travels through the tree from bottom to top deactivating the *empty* field if it is necessary.

If the semidynamic k-d tree is used during the multi-dimensional Scale Saliency algorithm, the scale-space must be explored from s_{max} to s_{min}. After estimating the entropy of a pixel $\mathbf{x}$ at scale s, the nodes that are not part of $R_{\mathbf{x}}$ at scale $s-1$ are removed prior to entropy estimation at that lower scale. Exploring the scale-space in reverse order while uprating the tree would be more efficient, but as stated before we are not aware of fully dynamic k-d tree algorithms that implement an insert operation.

4.2.2 Rényi α-entropy estimation

Rényi α-entropy [Rényi, 1961] is a generalization of Shannon's entropy, and therefore it is a measure of the uncertainty associated with a random variable. The Rényi α-entropy of a random variable X is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^\alpha \right) \quad , \tag{4.1}$$

where p_i are the probabilities of the random values $x_i \in X$. Hero and Michel introduced a estimator of the Rényi α-entropy from MSTs [Hero and Michel, 1999] or KNNGs [Costa and Hero, 2004]. The method is straightforward and it is based on the length of the entropic graph spanning the samples: higher spacing between samples means more

uncertainty, and as a consequence, higher entropy (see Fig. 4.2). In a d-dimensional space, with $d \geq 2$, the α-entropy estimator

$$H_\alpha(X_n) = \frac{d}{\gamma}\left[\ln \frac{L_\gamma(X_n)}{n^\alpha} - \ln \beta_{L_{\gamma,d}}\right] \tag{4.2}$$

is asymptotically unbiased and consistent with the pdf of the samples. In Eq. 4.2, γ depends on α and data dimensionality ($\alpha = (d-\gamma)/d$) and the bias correction $\beta_{L_{\gamma,d}}$ depends on the graph minimization criterion (that is, the type of graph construction – MST, KNNG, etc.), but it does not depend on the pdf. This bias can be approximated by (i) a Monte Carlo simulation from uniform random samples in the unit cube $[0,1]^d$ and (ii) an approximation for large d [Bertsimas and van Ryzin, 1990] given by

$$\beta_{L_{\gamma,d}} = \frac{\gamma}{2}\ln(\frac{d}{2\pi e}) \ . \tag{4.3}$$

We ignored this bias in our experiments. The value of $L_\gamma(X_n)$ is computed as the sum of the weighted length of the MST or KNNG edges $\{e\}$:

$$L_\gamma(X_n) = \min_{M(X_n)} \sum_{e \in M(X_n)} |e|^\gamma \ , \tag{4.4}$$

where $M(X_n)$ is the possible set of edges spanning the graph (the minimum $M(X_n)$ denotes the MST or the KNNG), $X_n = \{x_1, \ldots, x_n\}$ is the set of vertexes, $\{e\}$ is the set of edges, $|e|$ is the edge length and $0 < \gamma < d$.

Rényi α-entropy estimation from entropic graphs is suitable for $0 \leq \alpha < 1$. Its value converges to that of the Shannon's entropy when $\alpha \to 1$, but we can not set $\alpha = 1$ in Eq. 4.1. Peñalver *et al.* approximated the value of H_α for $\alpha = 1$ (H_1) by means of a continuous function that captures the tendency of H_α near that value [Peñalver et al., 2006][Peñalver et al., 2009]. Let α^* be the α value that best approximates H_1. This α^* is given by

$$\alpha^* = 1 - \frac{1.271 + 1.3912e^{-0.2488d}}{n} \ . \tag{4.5}$$

We did not apply Eq. 4.2 and Eq. 4.5 in our experiments. The reasons are:

- The approximation in Eq. 4.5 is obtained after experiments with Gaussian distributions and $2 \leq d \leq 5$. In general, multi-dimensional data sampled from an image does not follow a Gaussian distribution.

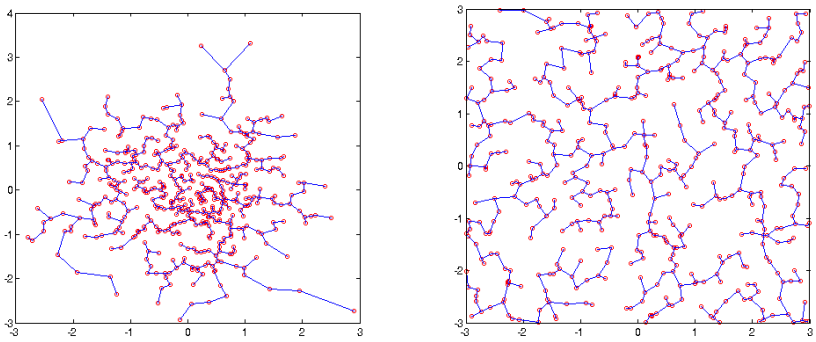


Figure 4.2: Two examples of entropy estimation from MSTs, using Eq. 4.2 and Eq. 4.5 ($\alpha^* = 0.9958$). Left: the samples are obtained from a Gaussian distribution ($H = 1.1838$). Right: the samples are obtained from an uniform distribution. Due to the fact that the samples are sparser, the length of the MST is larger ($H = 1.9762$).

Furthermore, we report experimental results for up to 31D data (see Sections 4.5.1 and 4.5.4).

- Our preliminary experiments showed that the computation of Eq. 4.2 is unstable in the context of multi-dimensional images when d is low and we use MSTs or KNNGs with a low value of k. The main cause of this instability is the presence of zero-length edges, that produce a graph for which $L_\gamma(X_n) < n^\alpha$ and as a consequence negative entropy values. A possible solution, inspired by how Neemuchwala *et al.* [Neemuchwala et al., 2006] solved the problem of the the α-mutual information instability, is to add uniform noise to the pixel values. However, this method affects the estimation.

4.2.3 Leonenko's entropy estimation

In this section we introduce the entropy estimation from k-nearest neighbor distances proposed by Leonenko *et al.* [Leonenko et al., 2008]. Their work is aimed to estimate the Rényi α-entropy [Rényi, 1961]

$$H_\alpha^* = \frac{1}{1-\alpha} \log \int f^\alpha(x) dx, \alpha \neq 1 \tag{4.6}$$

or the Havrda and Charvát entropy [Havrda and Charvát, 1967], also called Tsallis entropy [Tsallis, 1988]

$$H_\alpha = \frac{1}{\alpha - 1}\left(1 - \int f^\alpha(x)dx\right), \alpha \neq 1 \tag{4.7}$$

from N i.i.d. samples, based on the work by Kozachenko and Leonenko [Kozachenko and Leonenko, 1987]. The authors state that H_α^* is a strictly increasing concave function of H_α. Thus, the maximization of H_α and H_α^* are equivalent. When α tends to 1, both H_α and H_α^* tend to the Shannon's entropy:

$$H_1 = -\int f(x)\log f(x)dx \ . \tag{4.8}$$

The estimation of Eq. 4.8 from N i.i.d. samples is:

$$\hat{H}_{N,k} = \frac{1}{N}\sum_{i=1}^{N}\log \xi_{N,i,k} \ , \tag{4.9}$$

being

$$\xi_{N,i,k} = (N-1)e^{-\psi(k)}V_d(\rho_{k,N-1}^{(i)})^d \ . \tag{4.10}$$

In the latter equation, V_d is the volume of the d-dimensional unit ball, $\rho_{k,N-1}^{(i)}$ is the k-nearest neighbor of i when taking the rest of $N-1$ samples, and

$$\psi(z) = \frac{\Gamma'(z)}{\Gamma(z)} \tag{4.11}$$

is the digamma function. This function is $\psi(1) = -\gamma$, being $\gamma \simeq 0.5772$ the Euler constant, and $\psi(k) = -\gamma + A_{k-1}$ for an integer $k \geq 1$. Finally, $A_0 = 0$ and

$$A_j = \sum_{i=1}^{j}\frac{1}{i} \ . \tag{4.12}$$

The non-plug-in estimator in Eq. 4.9 requires to search for the k-nearest neighbors of each sample. In our entropic graphs based Scale Saliency algorithm we build the KNNG over the samples in $R_{\mathbf{x}}$ following the semidynamic method described in Sec. 4.2.1. This algorithm keeps the order

of the nearest neighbors and efficiently updates the KNNG in the scale-space, from s_{max} to s_{min}. An example is given in Fig. 4.3.

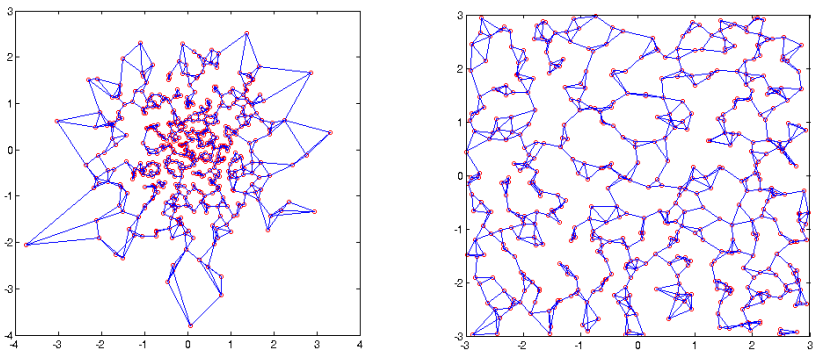


Figure 4.3: Two examples of entropy estimation from KNNGs, using Eq. 4.9 and $k = 3$. Left: the samples are obtained from a Gaussian distribution ($H = 2.7553$). Right: the samples are obtained from an uniform distribution. ($H = 3.6351$).

4.2.4 The Friedman-Rafsky test and the Henze-Penrose divergence

In the latter sections we have surveyed several entropy estimators that overcome the main drawback of histograms: their exponential spatial and temporal complexity with respect to data dimensionality. We discussed entropy estimation from entropic graphs, a non-plug-in method that does not require a previous estimation of the underlying pdf of the samples. However, there is another step in the Scale Saliency algorithm that depends on the data pdf: entropy peaks in the scale-space must be weighted by means of a self-dissimilarity measure between scales, which value must be in the range $[0, 1]$ (Eq. 2.78). This weight penalizes features that are salient over a wide range of scales (recall that the aim of the algorithm is to detect features that are salient only over a narrow range of scales). In the entropic graphs based Scale Saliency algorithm, we replaced this measure with an estimation of the Henze and Penrose divergence. The Henze and Penrose divergence [Henze and Penrose, 1999] between two distributions f and g is

$$D_{HP}(f||g) = \int \frac{p^2 f^2(z) + q^2 g^2(z)}{pf(z) + qg(z)} dz \ , \tag{4.13}$$

where $p \in [0,1]$ and $q = 1 - p$. This divergence is the limit of the Friedman-Rafsky run length statistic [Friedman and Rafsky, 1979], that in turn is a multi-dimensional generalization based on MSTs of the Wald-Wolfowitz test. The Wald-Wolfowitz statistic computes the divergence between two distributions f_X and f_O in $\mathcal{R}^d$, when $d = 1$, from two sets of n_x and n_o samples, respectively. First, the $n = n_x + n_o$ samples are ordered in ascending order and labeled as X and O according to their corresponding distribution. The test is based on the number of runs R, being a run a sequence of consecutive and equally labeled samples. The test is calculated as:

$$W = \frac{R - \frac{2n_o n_x}{n} - 1}{\left(\frac{2n_x n_o (2n_x n_o - n)}{n^2 (n-1)}\right)^{\frac{1}{2}}} \ . \tag{4.14}$$

The two distributions are considered similar if R is low and therefore W is also low. This test is consistent in the case that n_x/n_o is not close to 0 or ∞, and when $n_x, n_o \to \infty$. The Friedman-Rafsky test generalizes Eq. 4.14 to $d > 1$, due to the fact that the MST relates samples that are close in $\mathcal{R}^d$. Let $X = \{x_i\}$ and $O = \{o_i\}$ be two sets of samples drawn from f_X and f_O, respectively. The steps of the Friedman-Rafsky test are:

1. Build the MST over the samples from both X and O.
2. Remove the edges that do not connect a sample from X with a sample from O.
3. The proportion of non-removed edges converges to 1 minus the Henze Penrose divergence (Eq. 4.13) between f_X and f_O.

See an example in Fig. 4.4. The application of the Friedman-Rafsky test to the entropic graph based Scale Saliency algorithm is straightforward. Let s be the scale in which an entropy peak was found. In order to weight that entropy value, we must calculate the dissimilarity with respect to the scale $s - 1$ (Eq. 2.78). Let $M(X_{m,s})$ and $M(X_{n,s-1})$ be the entropic graphs used to estimate entropy at scales s and $s - 1$ (thus, $m > n$). The advantage of the

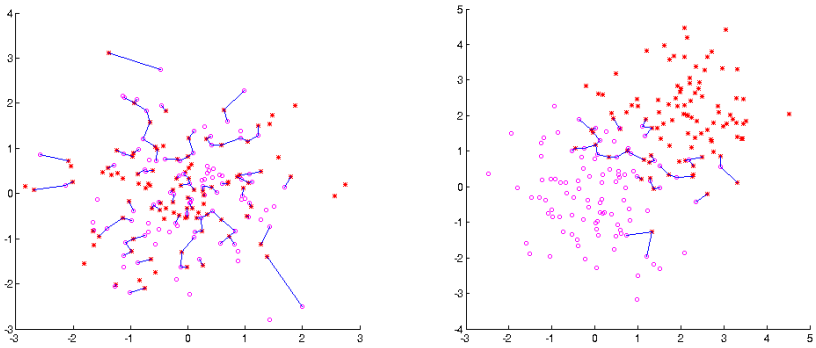


Figure 4.4: Two examples of Friedman-Rafsky estimation of the Henze and Penrose divergence applied to samples drawn from two Gaussian densities. Left: the two densities have the same mean and covariance matrix ($D_{HP}(O||X) = 0.5427$). Right: the two densities have different means ($D_{HP}(O||X) = 0.8191$).

Friedman-Rafsky test is that it is not necessary to to build an additional MST or KNNG since $X_{n,s-1} \subset X_{m,s}$ (new pixels are added to the previous ones as we increase the scale). Therefore, $M(X_{m,s})$ is the entropic graph that spans $X_{n,s-1} \bigcup X_{m,s}$. The weighting step only requires to count the amount of edges in $M(X_{m,s})$ that connect a node from $X_{m,s}/X_{n,s-1}$ to a node from $X_{n,s-1}$.

4.3 Entropy and divergence estimation based on the k-d partition algorithm

Our experiments (Section 4.5.1) demonstrate that the main drawback of the entropy estimation methods based on entropic graphs is the high temporal cost of building the underlying data structures. This computational burden is due to the calculation of distances. A new entropy estimator by Stowell and Plumbley overcomes this problem [Stowell and Plumbley, 2009]. They proposed an entropy estimation algorithm that relies on data spacing without computing any distance. This method is inspired by the data partition step in the k-d tree algorithm.

Let X be a d-dimensional random variable, and $f(x)$ its pdf. Let $A =$

$\{A_j | j = 1, \ldots, m\}$ be a partition of X for which $A_i \cap A_j = \emptyset$ if $i \neq j$ and $\bigcup_j A_j = X$. Then, we can approximate $f(x)$ in each cell as:

$$f_{A_j} = \frac{\int_{A_j} f(x)}{\mu(A_j)} \ , \tag{4.15}$$

where $\mu(A_j)$ is the d-dimensional volume of A_j. If $f(x)$ is unknown and we are given a set of samples $X = \{x_1, \ldots, x_n\}$ from it, being $x_i \in \mathcal{R}^d$, we can approximate the probability of $f(x)$ in each cell as $p_j = n_j/n$, where n_j is the number of samples in cell A_j. Thus,

$$\hat{f}_{A_j}(x) = \frac{n_j}{n\mu(A_j)} \ , \tag{4.16}$$

being $\hat{f}_{A_j}(x)$ a consistent estimator of $f(x)$ as $n \to \infty$ [Breiman et al., 1984][Zhao et al., 1990]. The differential entropy is defined as:

$$H = -\int f(x) \log f(x) dx \ . \tag{4.17}$$

We can plug Eq. 4.16 in the latter equation to obtain the entropy estimation for A:

$$\hat{H} = \sum_{j=1}^{m} \frac{n_j}{n} \log \left(\frac{n}{n_j} \mu(A_j) \right) \ . \tag{4.18}$$

The partition is created recursively following the data splitting method of the k-d tree algorithm. At each level, data is split at the median along one axis. Then, data splitting is recursively applied to each subspace until an uniformity stop criterion is satisfied. The aim of this stop criterion is to ensure that there is an uniform density in each cell in order to best approximate $f(x)$. The chosen uniformity test is fast and depends on the median. The distribution of the median of the samples in A_j tends to a normal distribution that can be standardized as:

$$Z_j = \sqrt{n_j} \frac{2med_d(A_j) - min_d(A_j) - max_d(A_j)}{max_d(A_j) - min_d(A_j)} \ , \tag{4.19}$$

where $med_d(A_j)$, $min_d(A_j)$ and $max_d(A_j)$ are the median, minimum and maximum, respectively, of the samples in cell A_j along dimension d. An

improbable value of Z_j, that is, $|Z_j| > 1.96$ (the 95% confidence threshold of a standard normal distribution) indicates significant deviation from uniformity. Non-uniform cells should be divided further. An additional heuristic is included in the algorithm in order to let the tree reach a minimum depth level: the uniformity test is not applied until there are less than $\sqrt{n}$ data points in each partition, that is, until the level

$$L_n = \left\lceil \frac{1}{2} \log_2(n) \right\rceil \tag{4.20}$$

is reached.

Finally we make to remarks about the algorithm. Firstly, the partitioning dimension is sequentially selected in each recursive call. Stowell and Plumbley state that in order to obtain a consistent estimation, the amount of required samples increases exponentially with data dimensionality. During the Scale Saliency algorithm this requirement is not satisfied. Choosing the dimensions with lower variance can minimize the effect of low sample density. This heuristic is commonly used in the k-d tree algorithm. However, as can be seen in Sec. 4.5.3, it decreases the performance of the k-d partition based Scale Saliency in the case of low data dimensionality.

Secondly, the partition algorithm summarized above can produce infinite-volume cells. Eq. 4.18 can not be applied to these cells. Stowell and Plumbley proposed to bound the volume of infinite cells to finite volume by means of the Maximum Likelihood Estimate of their hyper-rectangular support: the data support of a cell is estimated from the extrema of the data samples in it. This solution affects the quality of the entropy estimation in Eq. 4.18, but is less biased than, for instance, removing infinity support cells.

4.3.1 A new divergence measure based on the k-d partition algorithm

We use the Friedman-Rafsky test, introduced in Sec. 4.2.4, in order to weight entropy peaks during the graph-based multi-dimensional Scale Saliency algorithm. Not only it is simple and based on entropic graphs itself, but also it yields values in the range $[0, 1]$ (as in the case of the self-dissimilarity measure in the Kadir and Brady algorithm [Kadir and Brady, 2001]). Now that we have presented another

method to estimate entropy from a multi-dimensional distribution (whose complexity does not increase exponentially with data dimensionality), a new self-dissimilarity measure based on k-d partition should be defined. As far as we are aware, such divergence measure can not be found in the literature.

Our k-d partition based divergence measure follows the spirit of the total variation distance [Denuit and Bellegem, 2001], but may also be interpreted as a L1-norm distance. The total variation distance between two probability measures P and Q on a σ-algebra F[1] is given by:

$$sup\{|P(X) - Q(X)| : X \in F\} \ . \tag{4.21}$$

In the case of a finite alphabet, the total variation distance is

$$\delta(P, Q) = \frac{1}{2} \sum_{x} |P(x) - Q(x)| \ . \tag{4.22}$$

Let $f(x)$ and $g(x)$ be two distributions, from which we draw a set X of n_x samples and a set O of n_o samples, respectively. If we apply the partition scheme of the k-d partition algorithm to the set of samples $X \bigcup O$, the result is a partition A of $X \bigcup O$, being $A = \{A_j | j = 1, \ldots, p\}$ (see Section 4.3). In the case of $f(x)$, the probability of any cell A_j is given by

$$p(A_j) = \frac{n_{x,j}}{n_x} = p_j \ , \tag{4.23}$$

where $n_{x,j}$ is the number of samples of X in cell A_j. In the same way, in the case of $g(x)$ the probability of any cell A_j is given by

$$p(A_j) = \frac{n_{o,j}}{n_o} = q_j \ , \tag{4.24}$$

where $n_{o,j}$ is the number of samples of O in the cell A_j. Since the same partition A is applied to both sample sets, and considering the set of cells A_j a finite alphabet, we can compute the total variation distance between $f(x)$ and $g(x)$ as:

$$D(O||X) = \frac{1}{2} \sum_{j=1}^{p} |p_j - q_j| \ . \tag{4.25}$$

[1] A σ-algebra over a set X is a non-empty collection of subsets of X (including X itself) that is closed under complementation and countable unions of its members.

The latter distance can be use as a self-dissimilarity measure in the Scale Saliency algorithm since it satisfies $0 \leq D(O||X) \leq 1$. The minimum value $D(O||X) = 0$ is obtained when all the cells A_j contain the same proportion of samples from X and O. By the other hand, the maximum value $D(O||X) = 1$ is obtained when all the samples in any cell A_j belong to the same distribution. We show in Fig. 4.5 two examples of divergence estimation using Eq. 4.25.

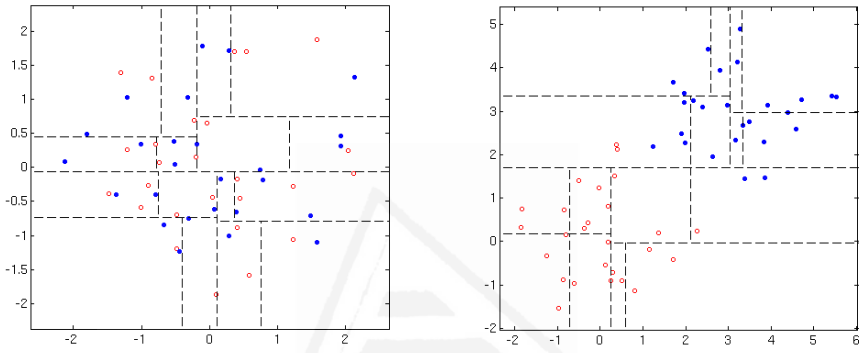


Figure 4.5: Two examples of divergence estimation applied to samples drawn from two Gaussian densities. Left: both densities have the same mean and covariance matrix ($D(O||X) = 0.24$). Right: the two densities have different means. Almost all the cells contain samples obtained from only one distribution ($(D(O||X) = 0.92$).

4.4 Multi-dimensional Scale Saliency: the algorithm

Our multi-dimensional Scale Saliency algorithm is shown in Algorithm 1. In order to generalize the Kadir and Brady implementation [Kadir and Brady, 2001], it makes use of the k-d partition entropy estimator and also our new divergence measure. The input of the algorithm is the set of m features extracted from each pixel in the image: gradient or color information (like for instance RGB values), the intensity values of a pixel through a video sequence or the output of a filter bank applied to each pixel, just to give some examples. The algorithm

is straightforward, and the only remarkable comment should be made about the interscale divergence (line 4 in Algorithm 1). Due to the fact that $X_{i-1} \subset X_i$, we may reuse the data partition performed during entropy estimation in order to calculate $D(X_{i-1}||X_i)$. The output of the algorithm is an array HW of saliency values that must be ordered; the most salient features of the image are those with highest saliency value.

Input: A m-dimensional array I containing m features for each pixel of the image
Output: An array HW containing weighted entropy values for all pixels on image at each scale
foreach *pixel* x *of image* **do**
  foreach *scale* s_i *between* s_{min} *and* s_{max} **do**
    (1) Create a m-dimensional sample set $X_i = \{\mathbf{x}_i\}$ from the local neighborhood of pixel $\mathbf{x}$ at scale s_i in I;
    (2) Apply k-d partition to X in order to estimate entropy $H(s_i)$
    if $i > s_{min} + 1$ **then**
      if $H(s_{i-2}) < H(s_{i-1}) > H(s_i)$ **then**
        `(* Entropy peak *)`
        (4) k-d partition divergence: $W = D(X_{i-1}||X_{i-2})$;
        (5) $HW(s_{i-1}, x) = H(s_{i-1}) \cdot W$;
      end
      else
        (6) $HW(s_{i-1}, x) = 0$;
      end
    end
  end
end

Algorithm 1: The multi-dimensional Scale Saliency algorithm.

Entropy and divergence estimation from entropic graphs may be also applied in this algorithm. Entropy and divergence are estimated from KNNGs or MSTs (see Sections 4.2.3 and 4.2.4). However, our experimental results in Section 4.5.1 demonstrate that this approach is remarkably slower than that based on k-d partition. Furthermore, in Section 4.5.3 we show that the quality of the extracted features is not improved. Entropy graphs have

not any advantage over the k-d partition algorithm.

4.5 Experimental results

In this section we test both estimation approaches: estimation based on entropic graphs and estimation based on the k-d partition method. These approaches are compared in terms of computational efficiency, quality of entropy and divergence estimation and quality and number of extracted features. These experiments were aimed to make a decision about the estimation method to be used in our multi-dimensional Scale Saliency algorithm (see Algorithm 1).

4.5.1 Computational time comparison

The purpose of this section is twofold: i) to compare the execution time of the MST and KNNG algorithms described in Section 4.2.1 in order to select a data structure for our graph-based multi-dimensional Scale Saliency algorithm and ii) to demonstrate that the complexity of our approach with respect to data dimensionality decreases when compared to the histogram-based Scale Saliency algorithm (Section 2.2.2). Furthermore, we also show that the practical complexity of the k-d partition method is remarkably lower than that of the KNNGs and MSTs based estimators.

We experimentally obtained the mean execution time per pixel of the MST and KNNG algorithms discussed in this chapter. An important conclusion of this experiment was that the fastest algorithm is not always that with the minimum expected theoretical complexity. Let n be the number of nodes, and let m be the number of edges. Due to the fact that we build MSTs and KNNGs over totally connected graphs we get $m = n^2$. The theoretical complexities are:

- The complexity of the Kruskal algorithm is $O(m \log m)$.
- The complexity of the Prim algorithm is $O(n \log n)$.
- The asymptotic complexity of the Katriel algorithm is $O(m + n \log n)$, that can be considered linear when $m >> n$. The expected complexity

is $mT + O(n \log n + \sqrt{mn})$, where T is the amount of time needed to check if an edge is removed.

- After building a k-d tree in $O(n \log n)$, the complexity of the KNNG algorithm is $O(\log n)$. This is the complexity of searching the k-nearest neighbors of any node.

In our experiment 100 pixels were randomly taken from each image in the *Bristol Hyperspectral Images Database*[2], a free dataset consisting on 29 hyperspectral images built from 31 spectrally filtered bands with a resolution of 256×256 pixels (see Fig. 4.6). Thus, any pixel is represented by a 31D vector. We set $s_{min} = 5$, and we computed the mean execution time per pixel for $s_{max} \in [8, 21]$ and $d \in [1, 31]$. We used the code provided by the authors of the Katriel algorithm[3]. Our Kruskal implementation was based on the Quicksort ordering algorithm and the Disjoint-Set data structure [Cormen et al., 2001]. Our Prim implementation relied on the Pairing Heap data structure [Fredman et al., 1986]. Regarding the KNNG algorithm, we set $k = 4$ for both the semidynamic [Bentley, 1990] and the non-dynamic algorithms. In the former case the data structure built at higher scales is reused at lower scales, remarkably decreasing the execution time. In the latter case, and in the case of all the MST algorithms, a new entropic graph is built for any pixel at any scale. The value $k = 4$ is a trade-off between execution time and estimation quality. Finally, it must be noted that we added uniform noise before building the MSTs (see Section 4.2.2). The results of the experiment are summarized in Fig. 4.7. As can be seen, the Kruskal algorithm was not included; its complexity strongly depends on edge density and our graphs are totally connected. As a consequence, its execution time was remarkably higher than that of the rest of algorithms. It should only be applied to sparse graphs or graphs having a small number of vertexes.

From all the tested entropic graphs algorithms the semidynamic KNNG is the fastest one. This was the expected outcome of the experiment since the semidynamic KNNG algorithm reuses its underlying data structure at different scales. This feature improves performance in the same way as cumulative histograms decrease execution time of the Kadir and Brady

[2]http://psy223.psy.bris.ac.uk/hyper

[3]http://www.mpi-inf.mpg.de/∼sanders/dfg/

Figure 4.6: Several example images from the *Bristol* dataset (RGB representation).

Scale Saliency algorithm (see Section 3.6.3). Katriel algorithm is faster than Prim algorithm, but their performances are closer. In both cases data dimensionality seems to not affect execution time as much as the range of scales. Although KNNG algorithms are more sensitive to data dimensionality, the range of scales is the most significant factor too. We finally chose the semidynamic algorithm to compare k-d partition and entropic graph approaches (see below).

Another consideration is the randomized nature of the Katriel algorithm. Recall that one step of the algorithm involves the selection of a random set of edges, that are used to discard other edges that are not likely to be part of the MST. Multiple applications of the algorithm to the same data can yield different MSTs, or even trees that are not a MST. This fact is illustrated in Fig. 4.8. It shows the mean error length after five executions of the Katriel algorithm for an increasing number of randomly generated 2D nodes. We define error length as the difference between the length of the MST obtained by the Katriel algorithm and the length of the MST obtained by any deterministic algorithm (Prim algorithm in this experiment). As can be seen, the mean error tends to zero as the number of nodes increase, remaining that value at 1200 nodes. The conclusion of this experiment is that the Katriel algorithm should be used in problems with a high amount of nodes. This is not the case of the Scale Saliency method. If $s_{max} = 20$, for instance, the maximum number of processed nodes during the algorithm is 1149.

Now we compare the execution time of our multi-dimensional Scale Saliency algorithm when using the semidynamic KNNG and the k-d partition methods. In this experiment we utilized the *Bristol Hyperspectral*

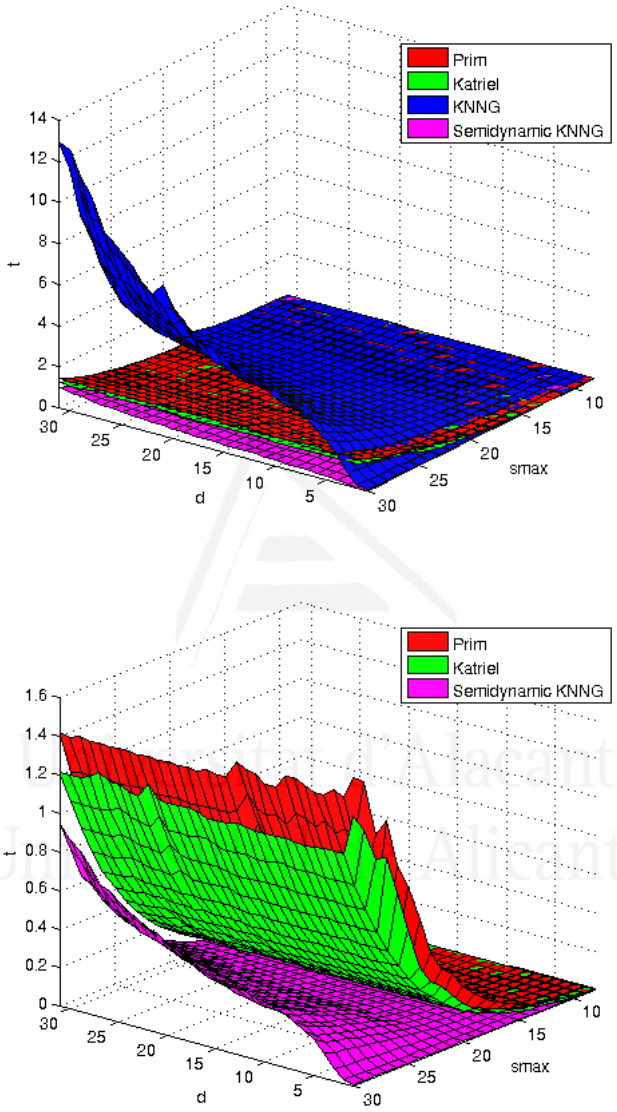


Figure 4.7: Mean execution time per pixel of the studied KNNG and MST algorithms. Top: results for all the algorithms. Bottom: results for all the algorithms except the non-semidynamic KNNG .

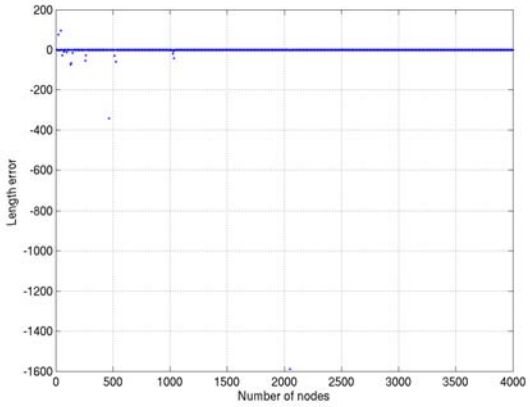


Figure 4.8: Mean length error of the Katriel algorithm for an increasing number of randomly generated 2D nodes.

Images Database (see above). The results demonstrate that our algorithm can cope with high data dimensionalities. Our approach can directly process these 31D images, but the experiment required the modification of the Scale Saliency code provided by Kadir and Brady[4], that is not prepared to process images formed by more than 3 bands. The results in Fig. 4.9 show the mean execution time of the different Scale Saliency approaches for the 29 images in the dataset as we added progressively from 1 to 31 image bands. In the case of the Kadir and Brady algorithm we performed histogram quantization, due to the exponential increase of its spatial complexity. This step is not necessary if Scale Saliency is based on KNNGs or the k-d partition estimator.

As expected, the execution time of the Kadir and Brady algorithm increases exponentially, due to the fact that its complexity is exponential with respect to data dimensionality. This is not the case of the multi-dimensional Scale Saliency. The expected complexity of the Scale Saliency method based on semidynamic k-d trees is $O(kn + n \log n)$ [Bentley, 1990], while the complexity of the Scale Saliency based on the k-d partition algorithm is $O(n \log n)$ [Stowell and Plumbley, 2009]. Their execution time increases almost linearly with data dimensionality. The remarkably lower execution time of the k-d partition based approach makes the multi-dimensional

[4]http://www.robots.ox.ac.uk/~timork/salscale.html

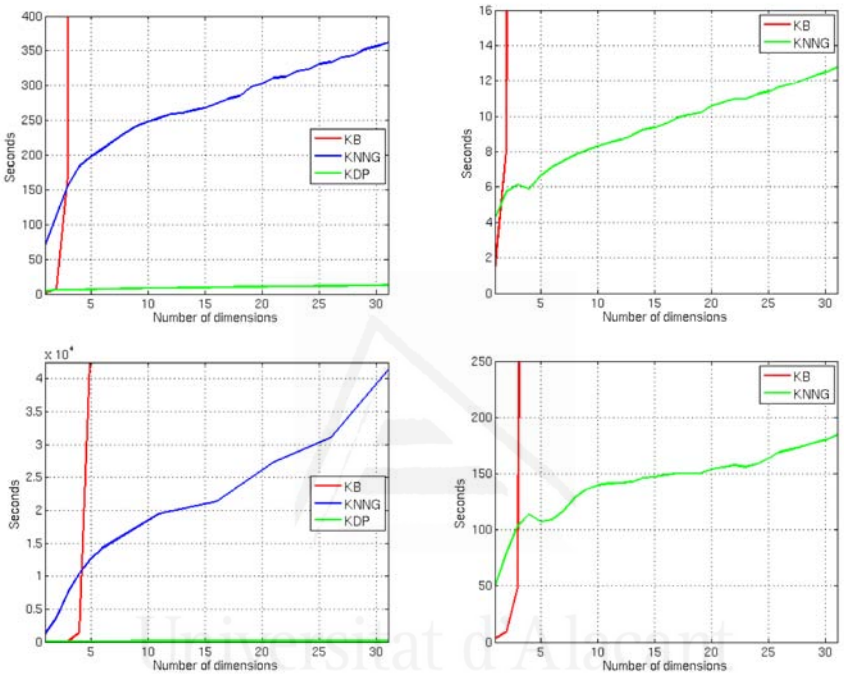


Figure 4.9: Comparison of the mean execution time between Kadir and Brady Scale Saliency (KB), KNNG based Scale Saliency (KNNG) and the k-d partition based Scale Saliency (KDP), as we increase the number of image bands from 1 to 31. Top row: $s_{min} = 5$, $s_{max} = 8$ and 64 histogram bins used in the Kadir and Brady algorithm. Bottom row: $s_{min} = 5$, $s_{max} = 20$ and 32 histogram bins used in the Kadir and Brady algorithm. In both cases the right column shows execution time of the k-d partition based method in detail, compared to that of the Kadir and Brady algorithm.

Scale Saliency algorithm feasible. It is impossible to apply the Kadir and Brady approach to high-dimensional data due to its temporal and spatial complexity. And although the KNNGs based approach solves this issue, its time requirements are still excessive. It must be noted that in Fig. 4.9 we compare results for the case of two different scale ranges. This parameter strongly affects the performance of the KNNG based Scale saliency algorithm. The k-d partition based Scale Saliency method seems to cope better with wider scale ranges, but the impact is still noticeable.

4.5.2 Quality of estimation

In this section we will test the estimation quality of the algorithms introduced in previous sections. We first compare the entropy estimator proposed by Leonenko *et al.* (Section 4.2.3) to the k-d partition estimator (Section 4.3), using Gaussian and uniform distributions. The normal distribution $N(\mu, \sigma^2)$ has the maximum entropy among all real-valued distributions with mean μ and standard deviation σ [Cover and Thomas, 1991]. The theoretical entropy of a Gaussian distribution in $\mathcal{R}^d$ with a covariance matrix Σ is given by

$$H_G = \frac{1}{2}\log((2\pi e)^d|\Sigma|) \ . \tag{4.26}$$

By the other hand, the uniform distribution on the interval $[a, b]$ is the maximum entropy distribution among all continuous distributions which are supported in the interval $[a, b]$ [Cover and Thomas, 1991]. The theoretical entropy value of an uniform distribution in $[a, b]$ is

$$H_U = \frac{1}{b-a}, a \leq b \ . \tag{4.27}$$

We measured the mean deviation (after 100 runs) from the theoretical entropy for both types of distributions as we increased data dimensionality and the number of pixels. This number of pixels corresponds to the amount of pixels in $R_{\mathbf{x}}$ for the scales between $s_{min} = 3$ and $s_{max} = 30$. The results are shown in Fig. 4.10. As one may expect, in all cases the estimation asymptotically improves as we increase the number of samples. Also, in all cases, increasing data dimensionality deteriorates entropy estimation.

None of the tested estimators performs better in all circumstances. The Leonenko *et al.* estimator approximates better the theoretical entropy of the Gaussian distribution, while the k-d partition estimator approximates better the theoretical entropy of the uniform distribution. It must be also noted that the Leonenko estimator does not require a high value of the parameter k; on the contrary, it yields better results for $k = 2$.

Despite these results, the Scale Saliency algorithm does not require an exact estimation of entropy, as long as the used entropy function approximates the trend of the Shannon's entropy as saliency increases. We performed an additional experiment in order to compare the trend of the the k-d partition entropy estimation to that of the the Leonenko *et al.* estimator. The experiment consisted in drawing N samples $x \in [0, 255]^d$ from a Gaussian (uniform) distribution, being N the number of pixels in $R_{\mathbf{x}}$ at $s_{max} = 30$. Then we computed the mean estimated entropy (over 100 runs) as we decreased the amount of samples, removing in each iteration the farthest sample from the center of mass of the set of samples. The results for different values of d are shown in Fig. 4.11 (Fig. 4.12). In these figures we also show the trend of the histogram-based estimation for $d = 2$ and $d = 3$ using 256 histogram bins.

For Gaussian data, the k-d partition algorithm approximates better the trend of the histogram based entropy estimation, even in the case of higher dimensions. From $d = 3$, the KNNG based estimation soon converges: it has less discriminative power. For uniform data both estimators soon reach and asymptote; however, the k-d partition curve still approximates better the shape of the histogram based estimation curve. This experiment demonstrates that k-d partition performs better in the case of high-dimensional data.

Now we validate our k-d partition based divergence, comparing its trend to that of the Friedman-Rafsky test. We conducted the experiment proposed by Neemuchwala in his Ph. D. thesis [Neemuchwala, 2004]: we compared the divergence of two sample sets (composed of 100 samples each) drawn from two Gaussian distributions, having both distributions the same covariance matrix as we increased the distance between Gaussian centers. We show the results of this experiment in Fig 4.13. In both cases the divergence (y axis) increases with the distance between Gaussian centers (x axis). The values of

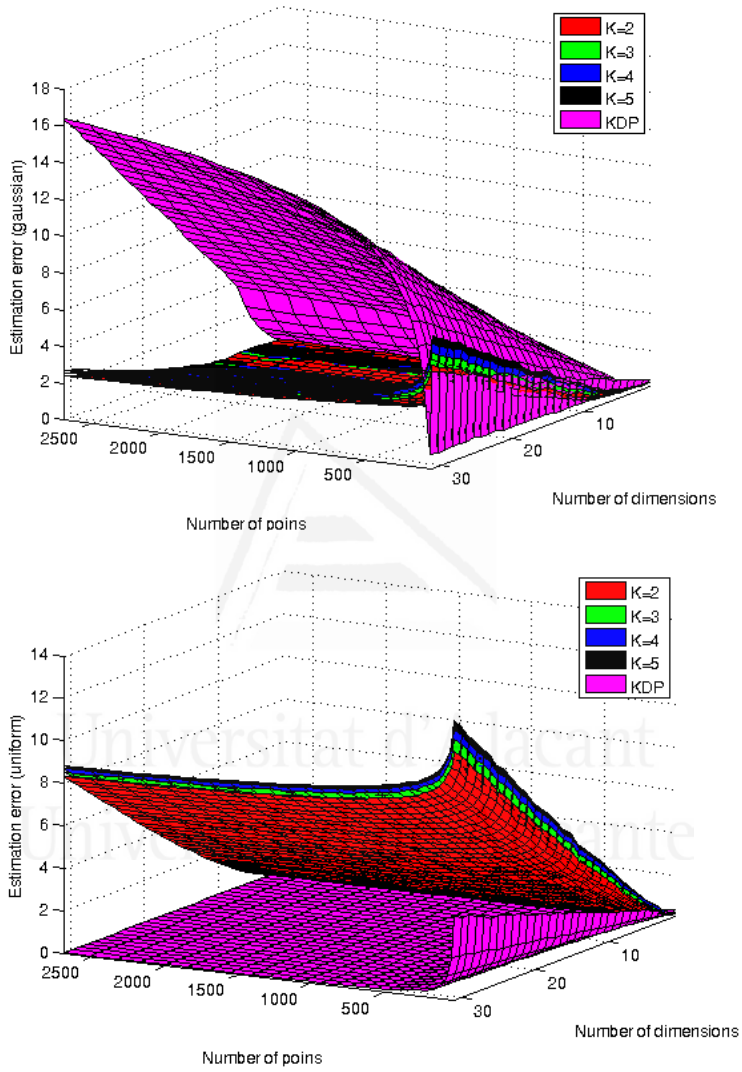


Figure 4.10: Deviation from the theoretical entropy of uniform (in the range $[-3, 3]^d$) and Gaussian (zero mean and $\Sigma = I$) distributions of the entropy estimated by the k-d partition method (KDP) and by the Leonenko *et al.* estimator for different values of k ($K = 2 \ldots 5$).

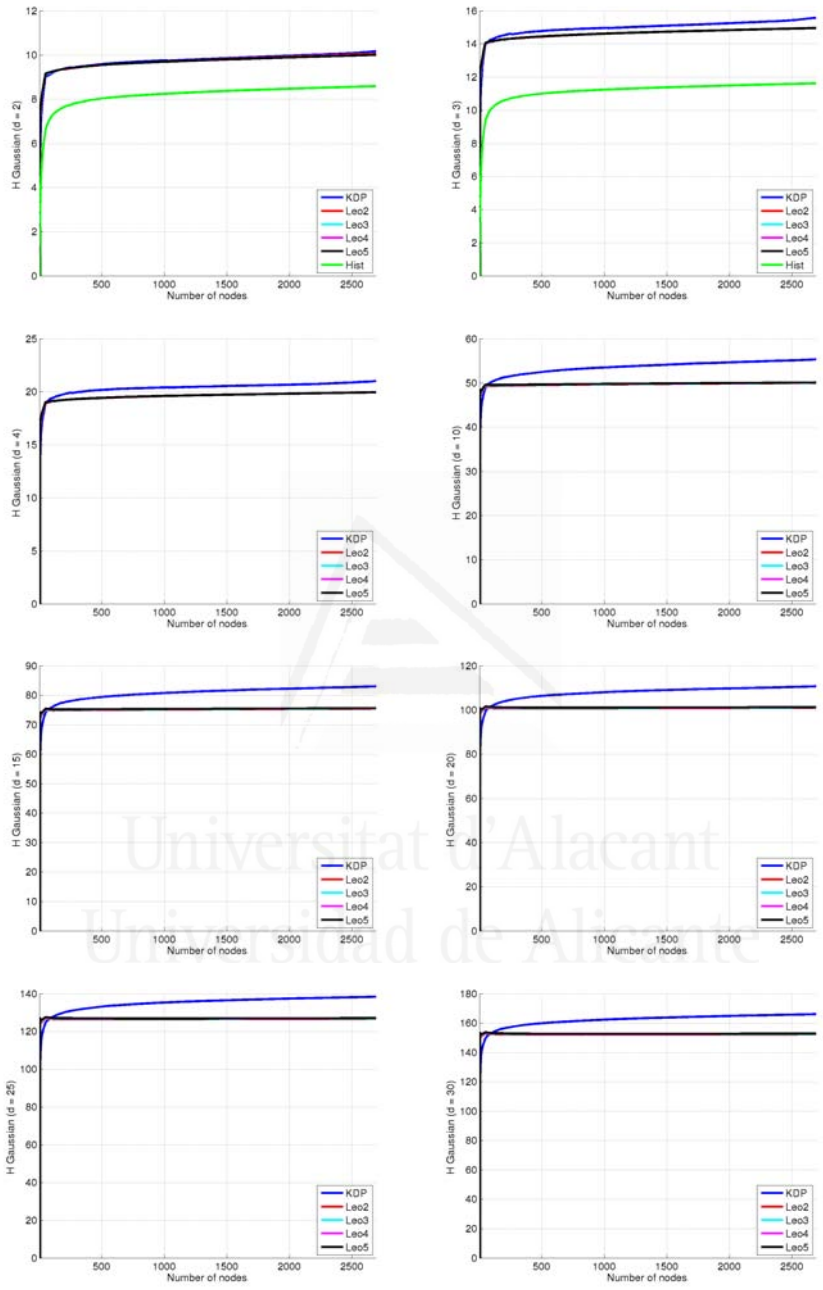


Figure 4.11: Entropy estimation from Gaussian data using histogram based estimation (Hist), Leonenko *et al.* estimation for different values of k (Leo2 to Leo5), and k-d partition estimation.

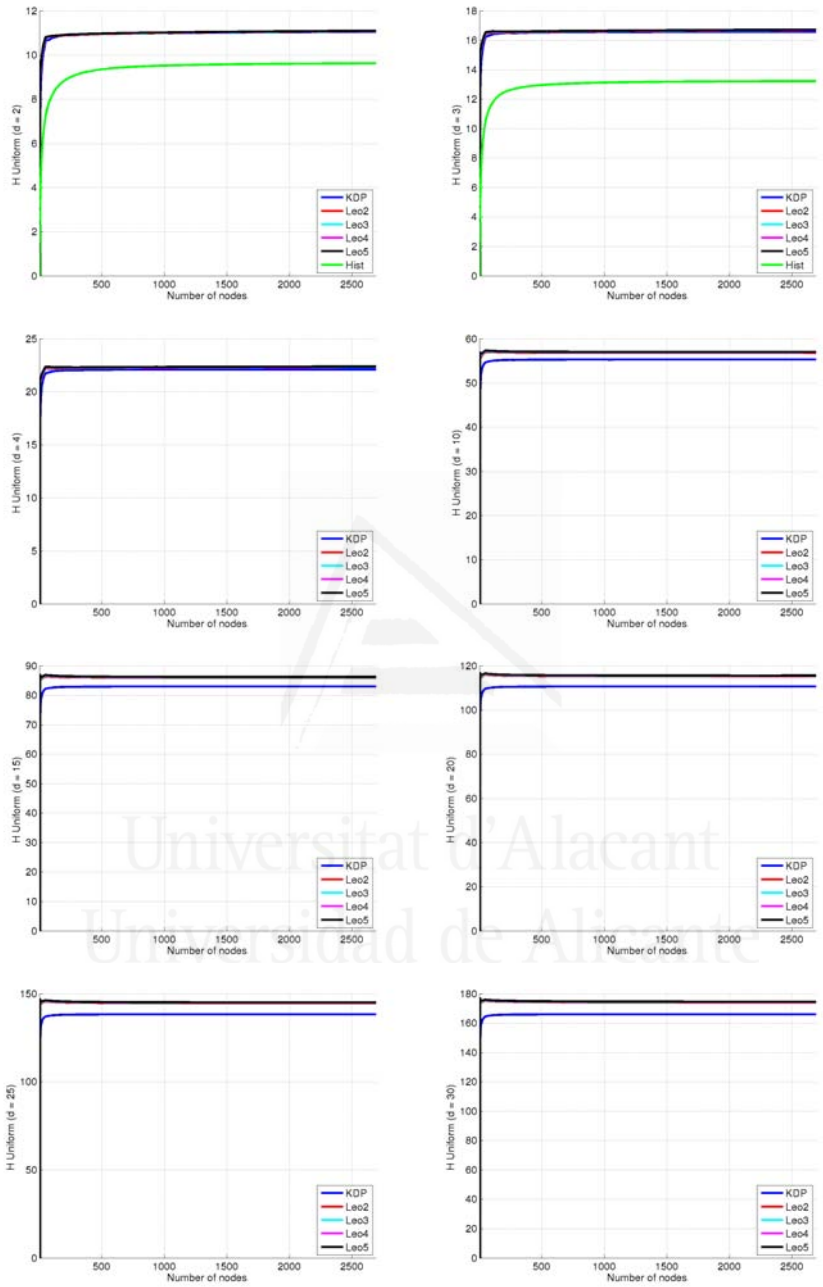


Figure 4.12: Entropy estimation from uniform data using histogram based estimation (Hist), Leonenko *et al.* estimation for different values of k (Leo2 to Leo5), and k-d partition estimation.

the Friedman-Rafsky test lie in the range $[0.5, 1]$. The range of values of our k-d partition based divergence depends on data dimensionality. Its trend is in all cases similar to that of the Friedman-Rafsky test, but its maximum value decreases as we increase d.

We draw two conclusions from these results. Firstly, although the k-d partition based Scale Saliency algorithm remarkably decreases execution time and can be applied to data for which processing by means of entropic graphs or histograms is unfeasible (in this thesis we report results for up to 31 dimensions), the divergence estimation is poorer as data dimensionality increases. Thus, our divergence should not be applied to too high-dimensional data. This limitation is not due to our divergence measure, but to the nature of the k-d partition algorithm, that requires at least 2^d samples in order to produce a consistent estimation for d-dimensional data. This requirement is not satisfied by the Scale Saliency algorithm in high dimensionalities: the number of pixels in $R_{\mathbf{x}}$ usually spans from less than 10 ($s = 3$) to less than 3000 ($s = 30$). And secondly, our divergence measure is more discriminative in cases of low data dimensionality. If $d < 25$, the range of values produced by our divergence is wider than that of the Friedman-Rafsky test.

4.5.3 Quality of the extracted features

In this section we will compare the results of our multi-dimensional Scale Saliency algorithm, using different approaches (estimation from entropic graphs and from the k-d partition method), to those of the Scale Saliency algorithm proposed by Kadir and Brady. We will first review the literature about evaluation of image feature extraction approaches. The study of this literature is important, not only because it will yield a general idea of the current state of the art of this type of evaluation techniques, but also because that it will justify the application of the measure that we use in this comparison: repeatability.

Evaluation of feature extraction algorithms

The paper about evaluation of interest point detectors written by Schmid *et al.* [Schmid et al., 2000] was a milestone in the history of this class

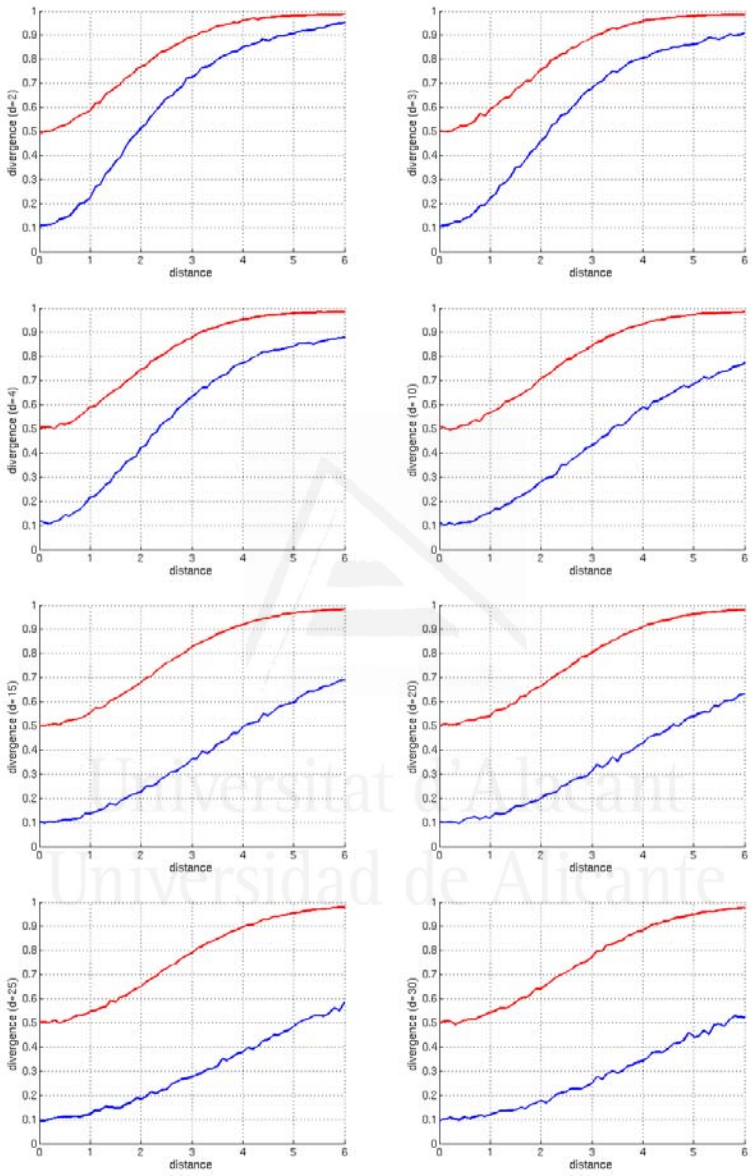


Figure 4.13: Comparison between the Friedman-Rafksy test (red) and our k-d partition based divergence (blue), for increasing data dimensionality and 100 samples per distribution.

of methods. Two measures are defined in this paper: repeatability and information content. The repeatability test measures the geometry stability of features, that is, their invariance under given image transformations (e.g. image scale and rotation or light variation). The information content test measures how distinguishable the features are, and it is based on the computation of a descriptor for each feature and on the comparison of these descriptors. If the extracted features are not distinguishable their descriptors will be similar.

The repeatability test described in [Schmid et al., 2000] can only be applied to planar scenes. Let I_1 be a reference image obtained from a given scene and let I_i be the result of applying a transformation to I_1. Let also x be a 3D point in the scene. The position of x in I_1 and I_i is defined by two projection matrices P_1 and P_i, respectively, being $x_1 = P_1 x$ and $x_i = P_i x$. Both projected points are related by means of a homography $x_i = H_{1i} x_1$, where $H_{1i} = P_i P_1^{-1}$. The matrix P_1^{-1} represents the back-projection of P_1, that only exists in the case of planar scenes.

Repeatability is defined as the ratio between the number of points detected in I_1 that are also detected in I_i and the total number or points. The test takes into account only those points that belong to the common part of the scene represented by both images. These points are:

$$\{\tilde{x}_1\} = \{x_1 | H_{1i} x_1 \in I_i\} \ , \tag{4.28}$$

$$\{\tilde{x}_i\} = \{x_i | H_{i1} x_i \in I_1\} \ . \tag{4.29}$$

The points x_1 detected in I_1 are not usually detected in the exactly corresponding location x_i in I_i. Thus, the correspondence criterion is relaxed by introducing a localization error ϵ. The set of points detected in I_i that are also detected in I_i is given by

$$R_i(\epsilon) = \{(\tilde{x}_1, \tilde{x}_i) | dist(H_{1i}\tilde{x}_1, \tilde{x}_i) < \epsilon\} \ . \tag{4.30}$$

Due to the fact that the amount of detected interest points in both images may be different, repeatability is defined as:

$$r_i(\epsilon) = \frac{|R_i(\epsilon)|}{\min(n_1, n_i)} \tag{4.31}$$

where $n_1 = |\{\tilde{x}_1\}|$, $n_i = |\{\tilde{x}_i\}|$ and $0 \leq r_i(\epsilon) \leq 1$.

Fraundorfer *et al.* [Fraundorfer and Bischof, 2004] introduced a new measure for the performance evaluation of image feature detection algorithms under changes in viewpoint in complex and non-planar scenes. This measure is based on trifocal geometry. Instead of computing the ground truth (the homography) from two images, three images are used. Trifocal geometry finds correspondences of points from two views of the scene to a third view.

Later on, Mikolajczyk *et al.* [Mikolajczyk et al., 2005b] generalized the repeatability test in order to assess the performance of affine-invariant feature extractors. The result of this kind of algorithms is not a set of interest points, but a set of interest regions (see Section 2.1.3). In this case, repeatability is defined as the relative amount of overlap between extracted features from a transformed and a reference image. Two features overlap if the overlap error is below a given threshold ϵ:

$$1 - \frac{R_{\mu_i} \bigcap R_{(H^T \mu_1 H)}}{(R_{\mu_i} \bigcup R_{(H^T \mu_1 H)})} < \epsilon \ , \tag{4.32}$$

where R_μ is the region defined by the parameters μ of the ellipse that represent the extracted affine-invariant feature, and H is the homography that relates the reference and the transformed image. Repeatability is the ratio between the number of overlapping regions and the minimum number of extracted regions in both images. Only those extracted regions belonging to common parts of the scene are taken into account. Once again, Fraundorfer *et al.* [Fraundorfer and Bischof, 2005] extended this repeatability measure to the non-planar case, by means of trifocal geometry. Mikolajczyk *et al.* also introduced a distinguishability test for affine-invariant features, based on the computation of descriptors for each extracted feature.

All these measures were applied in the context of image matching. Mikolajczyk *et al.* [Mikolajczyk et al., 2005a] stated that the evaluation results could be totally different in distinct contexts, like for instance in the context of image categorization. They proved their statement by evaluating the state-of-the-art feature extractors in this context. Feature extractors with good results in image matching had poor results in image class recognition, and vice versa.

We now enumerate the conclusions extracted from these works:

- In the context of image matching and for a low number of image features the best results are obtained by the MSER and IBR algorithms. By the other hand, in the case of a high number of image features the best results are obtained by the affine-invariant versions of the Harris and Hessian algorithms [Mikolajczyk et al., 2005b]. In general, the performance of the Scale Saliency algorithm is low.

- In simple 3D scenes MSER is also the algorithm that yields the best results. These results are remarkably better than that of the rest of algorithms. In complex 3D scenes, the performance of all tested algorithms was similar [Fraundorfer and Bischof, 2005].

- The evaluation of feature extractors in 3D scenes suggests that the repeatability and matching scores in real-world applications are much lower than first expected [Fraundorfer and Bischof, 2005]. Future work in the field of feature extraction from images should focus on this kind of scenes.

- In the case of image categorization the results were completely different. The worst classification results were obtained when using the MSER algorithm. The Scale Saliency method performed remarkably well in this context [Mikolajczyk et al., 2005a]. Furthermore, the feature extraction approach proposed by Kadir and Brady has been successfully applied to this task [Fergus et al., 2003].

- An interesting conclusion of the evaluation conducted by Mikolajczyk *et al.* [Mikolajczyk et al., 2005b] is that the image features extracted by different algorithms are complementary. Regions that are detected by a given kind of algorithms like the Harris and Hessian affine methods are not detected by other types of algorithms like for instance the MSER method, and vice versa. The error rate of Computer Vision tasks that rely on these features can be reduced if features extracted by means of different algorithms are used [Lazebnik et al., 2005][Sivic and Zisserman, 2003].

Our results

Now we compare the Scale Saliency algorithm proposed by Kadir and Brady to the entropic graphs and k-d partition based multi-dimensional Scale Saliency method. We will apply the repeatability test and we will use the code and the image dataset provided by Mikolajczyk *et al.* [Mikolajczyk et al., 2005b], that are freely available on the Web[5]. These code and images are commonly used in the feature extraction literature. We do not apply the information content test since its results may vary depending on the specific descriptor used [Mikolajczyk and Schmid, 2005].

The image dataset consists of several image sequences. Each image sequence is composed of a reference image and five transformed images, obtained after applying the same kind of transformation to the first one in increasing order of magnitude. The image sequences are: *graf* (viewpoint change in structured scene), *wall* (viewpoint change in textured scene), *boat* (scale and zoom change in structured scene), *bark* (scale and zoom change in textured scene), *bikes* (blur change in structured scene), *trees* (blur change in textured scene), *ubc* (JPEG compression) and *leuven* (light change). All these sequences contain color images, except the *boat* one, that we do not use in our repeatability experiments. In Fig. 4.14 we show the reference and a transformed image from each sequence.

We applied the three tested algorithms to all the images in the dataset. Then, for each algorithm and each image sequence we compute the repeatability between the reference image and the rest of images in that image sequence. The result is a plot for each image sequence that shows how the tested algorithms are affected by a particular type of image transformation. The performance of the algorithms is expected to decrease as the magnitude of the transformations increases. The resulting plots can be seen in Fig. 4.15. We tested the following algorithms: the Scale Saliency algorithm using grayscale intensity as input (*KB1D*), the Scale Saliency algorithm using color information as input (*KB3D*), our multi-dimensional algorithm based on Leonenko entropy estimation and the Friedman-Rafsky divergence test (both estimations are obtained from KNNGs), using color information as input (*Leo*), our multi-dimensional algorithm based on k-d partition entropy

[5] http://www.robots.ox.ac.uk/~vgg/research/affine/

Figure 4.14: The reference image and a transformed image from each sequence used in the repeatability experiment. From top to bottom: *graf*, *wall*, *bark*, *bikes*, *trees*, *ubc* and *leuven*.

estimation and our new k-d partition divergence using color information as input (*KDP*), and the KDP approach using color information as input and changing the order in which the dimensions are selected during the partition procedure (*KDPv*). In the k-d partition algorithm (*KDP*) the dimensions along which the data is partitioned are sequentially selected in each recursive call. In our *KDPv* version of the algorithm we select the maximum variance dimension in each recursive call. In all cases, the output was the set of the 1% most salient features extracted from the image.

In general, color information improves the repeatability of the extracted features. This is an expected result, since color information increases the distinguishability. The only exception is the *bark* sequence for which color information decreases the repeatability. We also expected that the results of the *KB3D* algorithm were better than those of the *Leo* and *KDP* approaches. In the case of the *Leo* algorithm, a small variation in the position of any node may produce a quite different KNNG, and as a consequence, it can affect entropy and divergence estimation. In the case of the *KDP* method infinite support cells are bounded to a finite volume, biasing the estimation [Stowell and Plumbley, 2009].

If we compare the *Leo* and *KDP* algorithms, we will see that any of them outperforms the other one in all cases. These results, along with the results of the computation time experiments (see Fig. 4.9), motivated us to base our final multi-dimensional approach on the k-d partition algorithm (see Section 4.4). The repeatability values of the *KDPv* algorithm were always lower than those of the *KDP* algorithm. It seems that for low data dimensionality (three dimensions in our experiments) data partition following a sequential dimension order provides higher-quality features. The worst results for our multi-dimensional approach were obtained for the *ubc* image sequence. The used estimators are sensitive to homogeneity of input data.

4.5.4 The effect of data dimensionality on the number of extracted features

The amount of detected salient regions may have an effect on the quality and the repeatability of a image feature extraction

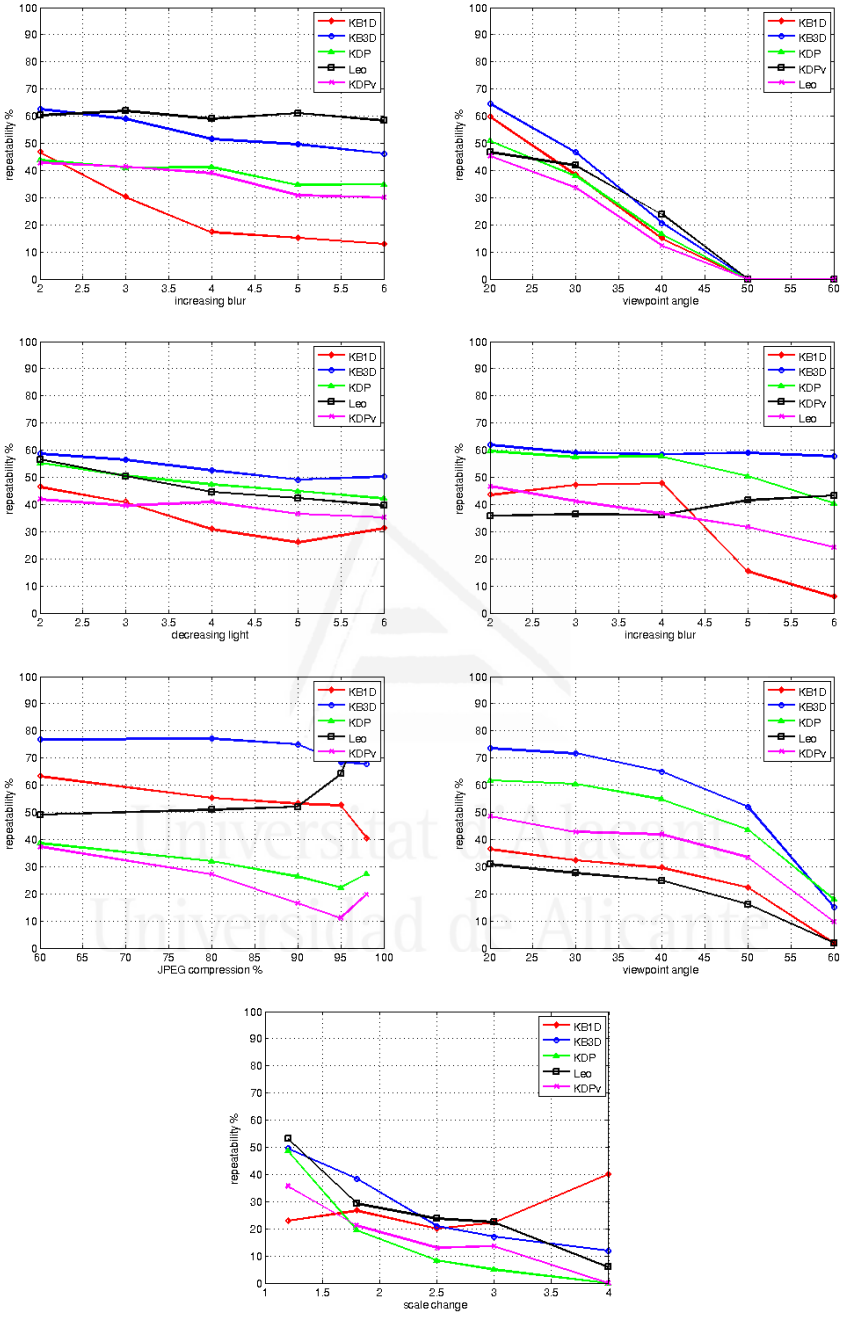


Figure 4.15: From left to right, and from top to bottom, repeatability plots for the *bikes*, *graf*, *leuven*, *trees*, *ubc*, *wall* and *bark* image sequences.

algorithm [Mikolajczyk et al., 2005b]. We may expect that different algorithms extract a different number of salient regions since these algorithms focus on different image properties. In the case of the multi-dimensional Scale Saliency algorithm we conducted an experiment in order to test if this factor is affected by the entropy estimation method applied (entropic graphs, k-d partition or histograms). We also wanted to test the effect of data dimensionality on the number of extracted features.

The final number of salient features extracted by the Scale Saliency algorithm depends on a parameter (a percentage of detected features). Recall that the algorithm selects a percentage of the weighted entropy peaks found during the entropy computation in the image and scale spaces in descending saliency order. However, this parameter can not be used to set the exact number of extracted salient features, due to the fact that several of these features are merged during the non-maximum suppression (feature clustering) step. Thus, instead of using the final number of extracted features in order to compare estimation methods, our experiments were based on the actual amount of entropic peaks detected during the algorithm.

In Fig. 4.16 we show the first results of our experiment. We computed the mean number of entropic peaks found by the multi-dimensional Scale Saliency algorithm when applied to the *Bristol* dataset images and as we increased the number of dimensions. In the plot we compare the results of the KNNGs based and k-d partition based Scale Saliency methods. When data dimensionality is low (below 5 dimensions), the k-d partition based approach outperforms the results of the Leonenko based one. It provides a remarkably higher amount of entropic peaks. For higher data dimensionalities the results of both methods are similar; however, the amount of entropic peaks detected by the k-d partition based algorithm is slightly higher. Both Scale Saliency methods are equivalent in terms of number of detected features, and could be equally used if only this factor is relevant.

In Fig. 4.17 we compare both multi-dimensional approaches to the Kadir and Brady Scale Saliency algorithm. It is unfeasible to apply this histogram based method to more than 4D data, so we can only show partial comparison results in Fig. 4.17. The amount of entropic peaks in this range of data dimensionalities (from 1 to 4) is remarkably higher in the case of estimation from histograms. This fact could be the cause of the better performance of

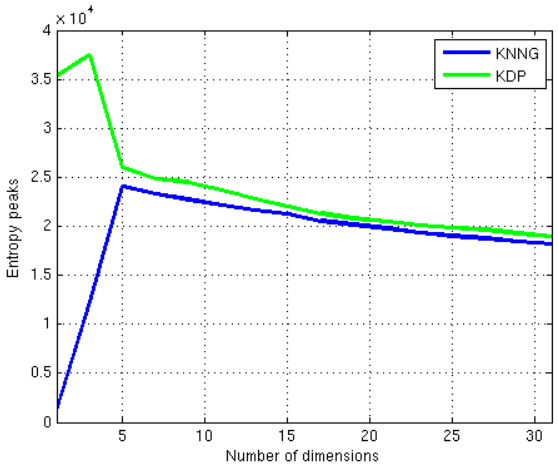


Figure 4.16: The effect of data dimensionality on the number of entropic peaks. This plot shows the mean number of entropic peaks detected by the entropic graph based Scale Saliency algorithm (KNNG) and the k-d partition based Scale Saliency algorithm (KDP) for the *Bristol* dataset images as we increase data dimensionality (from 1 to 31).

the Kadir and Brady Scale Saliency algorithm in the repeatability experiment (Section 4.5.3). In fact, the amount of salient regions detected by this algorithm was noticeably higher. And, as we previously stated, the number of detected features affects the results of the repeatability test.

It must be noted that the amount of detected entropy peaks decrease as we increase data dimensionality. We tried, for instance, to apply our multi-dimensional Scale Saliency algorithm to 128D data images, in which a SIFT descriptor [Lowe, 1999] was extracted for each image pixel, using a fixed scale. In most cases, no entropy peaks were detected by the algorithm. Our conclusion here is the same than the one drawn from the experiments in Section 4.5.2: there is a limit on the number of dimensions to which our multi-dimensional method can be applied.

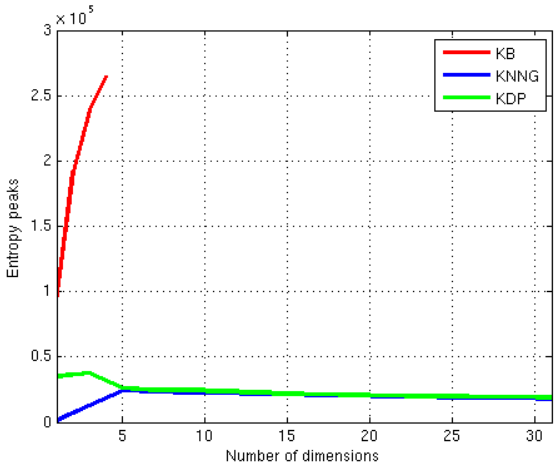


Figure 4.17: The results of the plot in Fig. 4.16 compared to the results of the Kadir and Brady Scale Saliency algorithm (KB).

4.6 Application: texture categorization

In this section we show the application of our algorithm to the texture categorization problem. Given an input image containing a single texture, the aim of texture categorization is to correctly indicate to which texture category or class it belongs. Our experiments in this section are based on the texture representation proposed by Lazebnik *et al.* [Lazebnik et al., 2005]. They represent texture images by means of a signature built from a set of features extracted from their grayscale intensities. Our hypothesis is that including additional data in the feature extraction process will increase the performance of categorization. In this section we first introduce the the algorithm by Lazebnik *et al.* and then we show our results.

4.6.1 A sparse texture representation

The main contribution of the texture representation proposed by Lazebnik *et al.* is the achievement of invariance to a wide range of image transformations, including viewpoint changes, by means of a texture signature based on image feature extraction. The signature building process is described as follows.

Firstly, features are extracted using the Harris affine algorithm [Mikolajczyk and Schmid, 2004b] and the Gårding and Lindeberg Laplacian blob detector [Gårding and Lindeberg, 1996] (see Chapter 2). Due to the fact that these methods focus on complementary features (the former searches for corners and the latter searches for blobs), better results are obtained when both are combined. The salient regions detected are represented as ellipses. In order to describe these regions they are first normalized, transforming them into unit circles.

Next, a descriptor is computed for each feature. Previous image categorization approaches require the estimation of the dominant orientation of each extracted feature in order to achieve rotation invariance. Lazebnik *et al.* state that orientation estimation tends to be unreliable, especially in the case of the Laplacian detector, which extracted features lack strong edges. Thus, they introduce two rotation invariant descriptors: the *intensity-domain spin images* and the *Rotation Invariant Feature Transform* (RIFT) [Lazebnik et al., 2005]. In both cases the descriptor can be computed without estimating the feature dominant orientation.

The **intensity-domain spin image** is a 2D histogram encoding the brightness distribution in the neighborhood of a pixel (the center of the detected salient region, in this case). The two dimensions of the histogram are the intensity i and the distance from the center d. The histogram is actually a soft histogram, in which each pixel contributes to more than one histogram bin. Given a pixel x, its contribution to the histogram bin (d, i) is given by

$$exp\left(-\frac{(|\mathbf{x}-\mathbf{x}_0|-d)^2}{2\alpha^2}-\frac{|I(\mathbf{x})-i|^2}{2\beta^2}\right) \tag{4.33}$$

where $\mathbf{x}_0$ is the center of the salient region, $I(\mathbf{x})$ the intensity of the pixel $\mathbf{x}$, and α and β are two parameters representing the soft width of the 2D histogram bin.

The **Rotation Invariant Feature Transform (RIFT)** descriptor is a rotationally invariant generalization of the SIFT descriptor proposed by Lowe [Lowe, 1999]. Firstly, the unit circle is split into concentric rings of equal width. A gradient orientation is computed within each ring, relative to the direction of the gradient in the center of the salient region. The result is a 2D histogram, in which one dimension represents the orientation θ and the other

one represents the distance from the center d (that is, the ring).

The next step in order to represent a texture image is to build its signature. The process starts by clustering the descriptors of the features extracted from that image, independently (that is, clustering is not applied to a set of images to build a *visual vocabulary*, like in the case of Bag of Words approaches [Sivic and Zisserman, 2003]). The method used is a simple agglomerative clustering: from an initial state, in which there are as many clusters as descriptors, the algorithm iteratively merges the closest pair of clusters, until a distance threshold is reached.

After clustering, the image is represented by a signature $\{(c_1, w_1), (c_2, w_2), \ldots, (c_k, w_k)\}$, where k is the number of clusters, c_i is the center of the cluster i and w_i the relative weight of the cluster i (i.e. the ratio between the number of descriptors in the cluster and the total number of descriptors in the image). This representation allows to compare textures by means of the *Earth Mover's Distance* (EMD) [Rubner et al., 2000]. The EMD between two signatures $S_1 = \{(c_1, w_1), (c_2, w_2), \ldots, (c_k, w_k)\}$ and $S_2 = \{(d_1, u_1), (d_2, u_2), \ldots, (d_k, u_k)\}$ is given by

$$d(S_1, S_2) = \frac{\sum_i \sum_j f_{ij} d(c_i, d_j)}{\sum_i \sum_j f_{ij}} \quad , \tag{4.34}$$

where the scalars f_{ij} are flow values that are determined by solving a linear programming problem, and $d(c_i, d_j)$ is the Euclidean distance between the cluster centers c_i and d_j. These centers are spin images or RIFT descriptors.

4.6.2 Multi-dimensional texture description

In the previous approach features are extracted from the graylevel intensity values of the image. In our experiments we built image signatures from higher dimensional data using our multi-dimensional Scale Saliency algorithm. We applied a Gabor filter bank to compute a multi-dimensional descriptor for each pixel in the image. Gabor filtering is a widely adopted technique for texture analysis [Bianconi and Fernández, 2007]. A 2D Gabor filter is defined as a harmonic function multiplied by a Gaussian function:

$$G(x, y) = exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad , \tag{4.35}$$

where $x' = xcos(\theta) + ysin(\theta)$ and $y' = -xsin\theta + ycos\theta$, and being λ the wavelength of the filter, θ its orientation, ψ the phase offset, σ the variance of the Gaussian kernel, and γ the spatial aspect ratio (the ellipticity). Our filter bank consists of 15 Gabor filters with different orientations and wavelengths that are applied to each pixel on the image. These filters are shown in Fig. 4.18. All filters have the same scale (that is determined by the value of σ), due to the fact that we obtained better results without modifying this parameter. After processing the texture image by means of the filter bank, we utilize multi-dimensional Scale Saliency in order to extract features from the resulting 15D data. The signature of the image is computed from this features following the method summarized above.

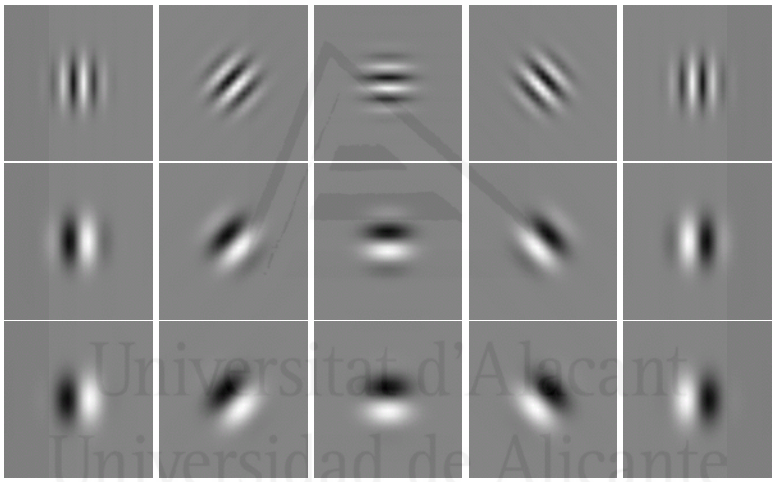


Figure 4.18: Gabor filter bank used in the texture categorization experiment.

4.6.3 Experimental results

We assess the performance of the multi-dimensional texture representation in the image retrieval task. In this experiment we use the *Brodatz* dataset[6]. This dataset contains 111 texture categories and 9 grayscale images per category making a total of 999 images. Any image comprises a single texture. It is a challenging dataset due to the fact that

[6]http://www.ux.uis.no/~tranden/brodatz.html

several texture categories are difficult to distinguish, even for a human observer. Several examples of these texture images are shown in Fig. 4.19.

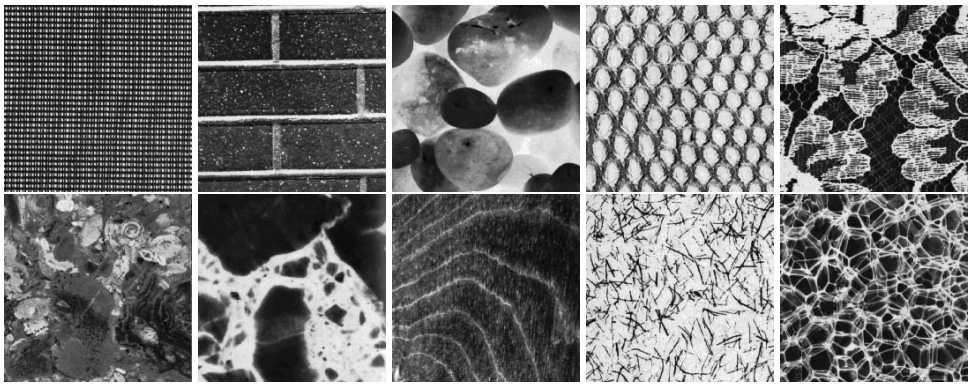


Figure 4.19: Several examples of texture images from the *Brodatz* dataset.

In the image retrieval experiment, all the images in the dataset are used once as query image. For each query image we select images from the database in increasing order of EMD. The result is a plot that shows the average recall of all the query images (being recall the ratio between the number of images from the query image class that were retrieved so far and the total number of images in that class) versus the number of closest images retrieved. In order to avoid the effect of variable amount of extracted features (recall that this parameter strongly affects the performance of feature extraction algorithms; see Section 4.5.4 and [Mikolajczyk et al., 2005b]), both Kadir and Brady Scale Saliency and multi-dimensional Scale Saliency were modified in order to always return 150 *clustered* features.

We show the results of the experiment in Fig. 4.20. We compared the performance of the grayscale Scale Saliency and k-d partition based multi-dimensional Scale Saliency using only RIFT, only spin images, and combining RIFT and spin images. In order to combine RIFT and spin images the total distance between two images is computed by adding the normalized EMDs calculated for each individual descriptor. As can be seen, multi-dimensional data yield better results in all cases. In Fig. 4.21 we also show several examples of application of the Kadir and Brady Scale Saliency and our multi-dimensional Scale Saliency to images in the Brodatz dataset.

Although multi-dimensional data increased the performance of the

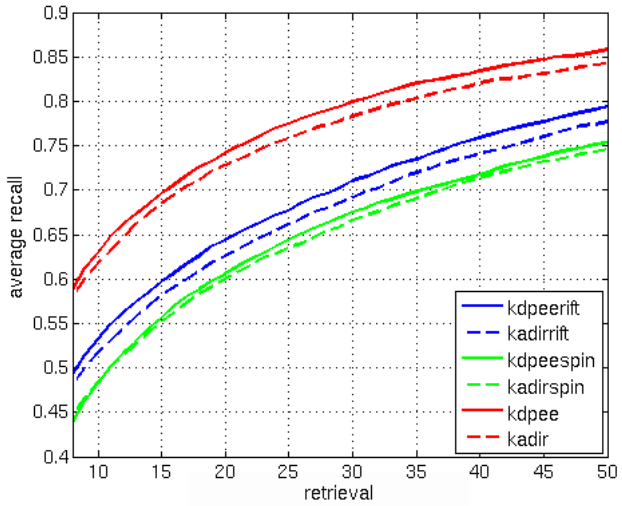


Figure 4.20: Average recall versus number of retrievals for Scale Saliency and RIFT (kadirrift), Scale Saliency and spin images (kadirspin), Scale Saliency and RIFT + spin images (kadir), multi-dimensional Scale Saliency and RIFT (kdpeerift), multi-dimensional Scale Saliency and spin (kdpeespin) and multi-dimensional Scale Saliency and RIFT + spin images (kdpee).

texture categorization task, its impact is not as noticeable as that of choosing an adequate descriptor. As can be seen in Fig. 4.20, the average recall is strongly affected by the type of descriptor used. The worst results were obtained for the case of spin images. RIFT increases the average recall, but the most significant improvement is achieved when combining both. Multi-dimensional data only yields a slight improvement over previous results. We strongly believe that our performance can be remarkably increased by selecting an optimal Gabor filter bank. This is a combinatorial problem that could be solved by means of feature selection techniques. This problem is out of the scope of this thesis and we leave it as future work.

4.7 Conclusions

In this chapter we have presented two different approaches to multi-dimensional Scale Saliency. The first one is based on the estimation

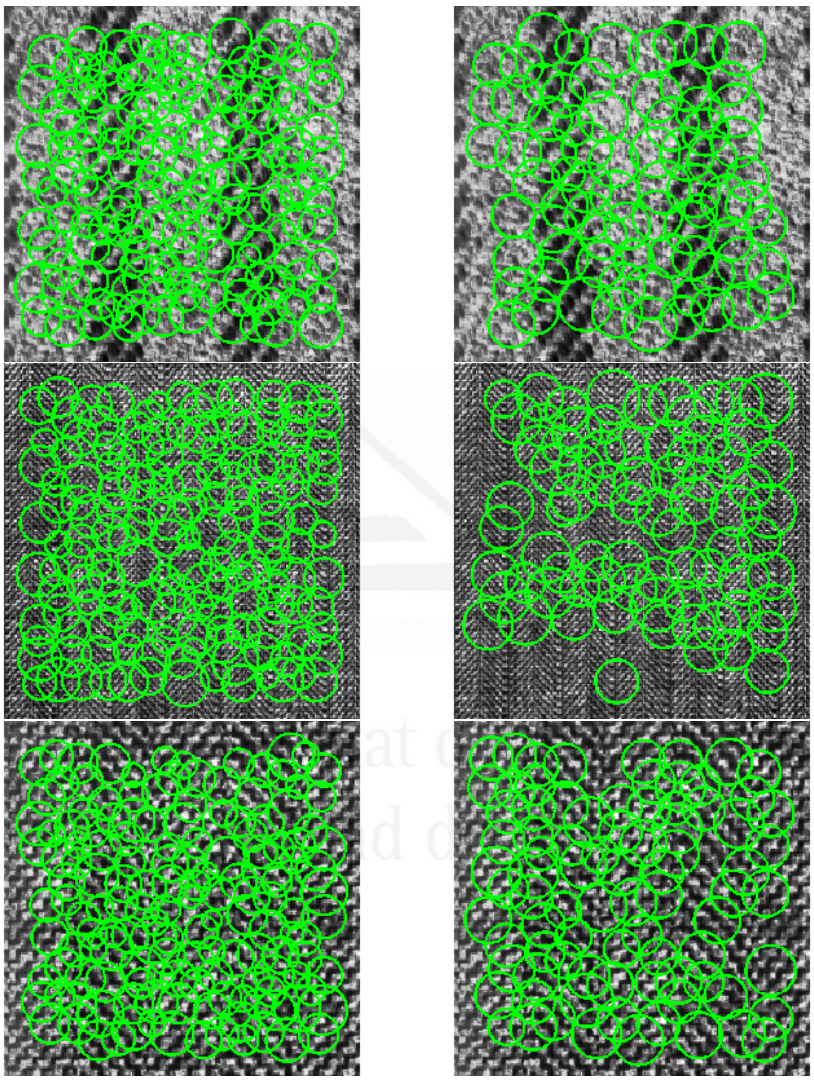


Figure 4.21: Examples of application of the multi-dimensional Scale Saliency (left) and the Scale Saliency (right) algorithms to three images in the *Brodatz* dataset.

of entropy and self-dissimilarity (divergence between scales) by means of Minimal Spanning Trees and K-Nearest Neighbor Graphs. The second one is based on the estimation of entropy and self-dissimilarity by means of the k-d partition algorithm. We analyzed both approaches. We also applied our multi-dimensional Scale Saliency algorithm to the texture categorization problem. The main contributions of this chapter are:

- We studied the application of alternative entropy and divergence estimators based on Minimal Spanning Trees and K-Nearest Neighbor Graphs to the multi-dimensional Scale Saliency algorithm: Shannon's entropy estimation from Rényi α-entropy (the main drawbacks of this method are its instability and that it was derived for Gaussian distributions), Kozachenko and Leonenko entropy estimation and the Friedman-Rafsky test, that is an estimation of the Henze-Penrose divergence.

- We studied the application of k-d partition algorithm, a fast entropy estimator based on data partition, to the multi-dimensional Scale Saliency algorithm.

- We introduced a new divergence measure, based on the k-d partition algorithm and the total variation distance, in order to estimate self-dissimilarity between scales. Our experiments demonstrated that the results of this divergence are comparable to those of the Friedman-Rafsky test. Furthermore, the range of values of our estimator is wider than that of the Friedman-Rafsky test in the case of low data dimensionality.

- We experimentally analyzed all the methods presented in the chapter in terms of computational efficiency, validity of estimation and quality and number of extracted features. Firstly, we showed that our multi-dimensional Scale Saliency approach reduces computational complexity with respect to data dimensionality from exponential to linear. The method based on k-d partition is remarkably more efficient than the one based on graphs. Both methods are equivalent in terms of quality of estimation and quality of the extracted features. Finally we showed that both approaches yield a similar amount of extracted

salient features when data dimensionality is higher than 5. For lower dimensionalities the k-d partition method outperforms the graph based one. However, both approaches extract a noticeable low number of features when compared to the Kadir and Brady Scale Saliency algorithm.

- A practical application of multi-dimensional Scale Saliency was presented. We demonstrated that multi-dimensional data can improve the performance of the texture categorization task when using the texture representation method by Lazebnik *et al.* [Lazebnik et al., 2005].

We reported results for our multi-dimensional Scale Saliency algorithm for data with up to 31 dimensions. The original Kadir and Brady method can not cope with such data dimensionality. Furthermore, using the k-d partition entropy estimator and the k-d partition based divergence we can process a 256×256 31D image in less than four minutes in a standard computer. However, the range of dimensions for which Scale Saliency can be applied is still limited. As we increase the number of dimensions, the performance of the k-d partition based divergence degrades (the range of divergence values is narrower) and the number of extracted features decreases.

Future work in multi-dimensional Scale Saliency includes to devise new applications of the algorithm. Examples of possible applications are: video analysis (each frame in a video sequence could represent a band in a multi-dimensional image), image categorization (how does additional information, like the output of a bank of filters or gradient magnitudes and orientations, affect the results of this type of applications?) and of course hyperspectral image analysis. In the texture categorization context we should also study the impact of using different Gabor filter banks, or even different input data. As we stated in Section 4.6.3, this is a combinatorial problem that may be treated by means of Machine Learning methods like feature selection.

Chapter 5

Conclusions

In this last chapter we firstly present a detailed summary of the contributions of this thesis. Then we discuss the limitations of the proposed methods and algorithms. Finally, we identify possible directions of future work.

5.1 Thesis summary

The work presented in this thesis was carried out in the context of image feature extraction. Image feature extraction algorithms detect high distinguishability regions. These distinguishable regions are the basic input of many high-level vision systems (e.g. object recognition or robot localization systems). Thus, we first presented an in-deep survey of the evolution and state of the art of image feature extraction algorithms, from the simple corner detectors that appeared in the seventies (we also summarized recent corner detection approaches), to the state-of-the-art affine-invariant feature extractors that detect image features that are robust to viewpoint changes. One of the highlights of this survey is the description of the scale-space representation.

We included in our survey a detailed description of the Scale Saliency algorithm proposed by Kadir and Brady. The Scale Saliency algorithm is the object of research in this thesis. It relies on Information Theory (Shannon's entropy) in order to estimate local saliency in an image and to detect image regions that are locally and highly distinguishable. It is an interesting

algorithm, mainly due to two reasons: i) using Information Theory is consistent with the objective of finding highly informative regions, and ii) its performance in object categorization problems was demonstrated to be high. However, the Scale Saliency algorithm suffers from several limitations. The most remarkable one is that this algorithm is the slowest one when compared to the rest of the state-of-the-art image feature extraction algorithms. Furthermore, its computational complexity increases exponentially with data dimensionality. The two main contributions of this thesis are aimed to address these issues.

Our first main contribution is a Bayesian filtering method to remove non-interesting points before applying the Scale Saliency algorithm. As a consequence, the execution time of the complete process is remarkably lower. Our initial hypothesis was that if an image region is homogeneous or non-salient at higher scales, it probably would also be homogeneous or non-salient at lower scales. After a previous empirical analysis of the evolution of the entropy function in the scale-space we performed a further statistical analysis. This statistical analysis showed that the entropy at the maximum scale can put an upper bound on the entropy value in the scale-space. Based on this result we proposed a first filtering approach in which we remove low-salient image regions at the maximum scale before applying the Scale Saliency algorithm to the rest of the image. This method is based on setting a normalized entropy threshold.

The main problem of this simple filtering approach is how to choose a threshold that may be applied to a set of images. We introduced a learning algorithm that sets the value of the entropy threshold for a given image category. This algorithm is based on Information Theory and the edge detection approach proposed by Konishi *et al.* Firstly, given a set of training images belonging to a same image category, we estimate two probability distributions that indicate the probability that a point is part or not of the most salient regions of an image, depending on its entropy value at the highest scale. Then, we can apply the Chernoff Information measure in order to determine if these two distributions are separable enough so a valid threshold can be learned. Next, we compute a range of valid threshold values for that image category from the Kullback-Leibler divergence between these two distributions. If we select a low value in this range, the probability of

removing actual salient regions is low, but the effect on execution time is also low. By the other hand, taking the maximum possible threshold value in this range makes the algorithm faster, but it is also possible that salient features are filtered. As a fact of interest, the Chernoff Information value and the width of the range of thresholds are related.

We conducted a series of experiments using two different image datasets in order to demonstrate that our filtering approach noticeable decreases execution time, while not strongly affecting the Scale Saliency algorithm results. We also studied the effect of several parameters like the amount of training images used, the number of extracted salient regions, or the range of scales. We also showed a practical application of our filtering method in the field of robot localization.

Our second main contribution in this thesis is the assessment of alternative entropy and divergence estimation methods with the aim of decreasing the complexity order of the Scale Saliency algorithm with respect to data dimensionality. Two types of approaches are studied: entropic graphs based estimation (Minimal Spanning Trees and K-Nearest Neighbor Graphs) and the k-d partition approach proposed by Stowell *et al.* We also presented a new divergence measure inspired by the total variation distance and based on the k-d partition algorithm. Our experiments with a hyperspectral image dataset (consisting on 31 band images) demonstrated that we achieve an almost linear complexity with respect to data dimensionality, when we use these estimators in the Scale Saliency algorithm. However, methods based on entropic graphs are slower and only the k-d partition algorithm allows to implement a feasible multi-dimensional Scale Saliency algorithm. Our experiments also compared these different entropy and divergence estimation methods in order to test their validity and the quality of the extracted features. The repeatability test showed that the performance of our approach in the case of color images is lower than that of the Kadir and Brady method (that can not be applied to higher-dimensional data), but it is higher than that of the grayscale based Scale Saliency.

Finally, we applied our multi-dimensional Scale Saliency algorithm to the texture categorization problem. Following the texture representation work by Lazebnik *et al.*, we describe the images in the Brodatz dataset from a set of features extracted from graylevel intensities and also from 15D information

obtained after applying a Gabor filter bank to all pixels in the image. Our experiments showed that multi-dimensional information improved the performance of the texture categorization.

5.2 Future work

The contributions in this thesis suppose an improvement of the Scale Saliency algorithm proposed by Kadir and Brady. However, we identified several limitations that should be subject of further analysis. We also suggest new lines of research arising from our work. Firstly, we should further analyze the role of the Chernoff Information measure in our Bayesian filtering approach. We stated that there exists a relationship between this measure and the learned range of valid thresholds, but we did not perform a deep study of this relationship.

We apply the Chernoff Information measure in order to test the "homogeneity" of an image category, that is, to test if a set of images is homogeneous enough so a valid threshold can be learned for this set. This idea is similar to the concept of intra-class variability, a measure used along with inter-class variability in several clustering algorithms in order to achieve an appropriate data separation. In our experiments images were manually grouped into categories. This image categorization criterion may not be optimal. Unsupervised clustering of images could provide a different separation in categories that increases the performance of the filtering algorithm. In this sense Chernoff Information can be applied as an intra-class variability measure.

Our Bayesian filtering approach is not based on the affine-invariant version of the Scale Saliency algorithm. The achievement of affine invariance by means of the extraction of anisotropic regions (ellipses) instead of isotropic regions (circles) was reported to increase performance. In the affine-invariant Scale Saliency approach two new parameters, apart from scale, are introduced. We could extend our statistical analysis of the entropy function in order to consider them. Another question to answer is if this method can be applied to different feature extractors, like the Harris affine (studying the cornerness function) or the Hessian affine algorithms.

Further improvement of the computational efficiency of the Scale Saliency method could be obtained by implementing additional filters. These filters may be organized as a filter cascade, in which the output of a filter could become the input of the next one, leaving the most complex ones for the last stages of the cascade. This idea of a filter cascade has been successfully applied in the past. Two examples are the face detection approach proposed by Viola and Jones or the text detection algorithm system implemented by Yuille *et al.*

Now let's focus on our multi-dimensional Scale Saliency algorithm. This algorithm overcomes an important limitation of the original method proposed by Kadir and Brady, and opens its application to a new range of problems. However, data dimensionality is still a limiting factor. Our k-d partition based divergence measure degrades as data dimensionality increases. Furthermore, the amount of detected entropy peaks also decrease with data dimensionality. We should analyze this data dimensionality limit. And there is an interesting question here: why the number of entropy peaks increases with data dimensionality if entropy is estimated by means of histograms and decreases in the case of entropic graphs or k-d partition based estimation?

Finally, our texture categorization experiments yield promising results. Regions of interest extracted from multi-dimensional data seem to improve the performance of the higher-level vision tasks that rely on these regions. Machine Learning techniques, like feature selection, may be applied in order to search for different Gabor filter banks, or even different input data, with the aim to increase the texture categorization performance. We can think of different vision applications for which feature extraction based on multi-dimensional data could be useful: video analysis, hyperspectral imaging, and so on.

Appendix A

Scientific production

This appendix is aimed to show the scientific results derived from this thesis.

A.1 Publications

Pablo Suau, Francisco Escolano, Exploiting Information Theory for Filtering the Kadir Scale-Saliency Detector, Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis (IBPRIA 2007), Lecture Notes in Computer Science, vol. 4478, pp. 146–153, Girona (Spain), 2007.

In this paper we introduced the algorithm described in Chapter 3. Information Theory (Chernoff Information and Kullback-Leibler divergence) is applied to learn a filtering threshold for an image category. Before extracting features from a new image belonging to that category, our filtering algorithm uses that threshold in order to remove points of the image that are non-salient at the highest scale. The experiments demonstrate that our approach remarkably decreases the computation time of the Scale Saliency algorithm, without strongly affecting final results.

Francisco Escolano, Boyán Bonev, Pablo Suau, Wendy Aguilar, Yann Frauel, Juan Manuel Sáez, Miguel Ángel Cazorla, Contextual Visual Localization: Cascaded Submap Classification, Optimized Saliency Detection, and Fast View Matching, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), pp. 1715–1722, San Diego, California (USA), 2007.

In this paper our Bayesian filtering approach is applied to the robot localization problem, as part of a system that is built using different techniques developed along with the rest of authors (see Section 3.7). Given a query image, captured by a camera mounted on a mobile robot, our system searches for the most similar image in a database in order to predict the actual localization of the robot in a given (indoor or outdoor) environment. The database was built from images belonging to a global environment that was divided into six different submaps. A different threshold was learned for each of these submaps. The output of a classifier that relies on simple filters determines in which of the six submaps the robot is located. Then, the query image is compared to a set of candidate images belonging to that submap in the database, by means of feature matching using Scale Saliency, SIFT descriptors, and a graph based algorithm that removes inconsistent matches. Scale Saliency is applied after removing non-interesting points using our Bayesian filtering method and the threshold corresponding to the detected submap.

Pablo Suau, Francisco Escolano, Bayesian Optimization of the Scale Saliency Filter, Image and Vision Computing, 26(9), pp. 1207–1218, 2008.

This paper provides additional background information about our Bayesian filtering algorithm, and also shows additional experimental results. An analysis of evolution of the entropy function in the scale-space demonstrates its smoothness, and also that entropy at the highest scale puts an upper bound on the entropy of a pixel in the selected range of scales. This result is key to the definition of our Bayesian filtering algorithm: image

regions that are non-salient at the highest scale will probably not be salient in the rest of scales. Our experiments show how our algorithm is affected by several parameters, like the number of extracted features and the range of scales. Finally, an application of the algorithm is discussed. This part of the paper is a summary of the previous robot localization paper in this list.

> Boyán Bonev, Francisco Escolano, Miguel Ángel Lozano, Pablo Suau, Miguel Ángel Cazorla, Wendy Aguilar, Constellations and the Unsupervised Learning of Graphs, Proceedings of the 6th IAPR Workshop on Graph-Based Representations in Patter Recognition (GBR2007), pp. 340–350, Alicante (Spain), 2007.

> Miguel Ángel Lozano, Francisco Escolano, Boyán Bonev, Pablo Suau, Wendy Aguilar, Juan Manuel Sáez, Miguel Ángel Cazorla, Region and constellations based categorization of images with unsupervised graph learning, Image and Vision Computing, 27(7), pp. 960–978, 2009

These papers describe an extended version of our previous robot localization system that was improved by means of structural information provided by graph matching algorithms [Bonev et al., 2007b]. The input of this kind of algorithms is a set of interest regions extracted by means of our filtered Scale Saliency method.

> Pablo Suau, Francisco Escolano, Multi-dimensional Scale Saliency Feature Extraction Based on Entropic Graphs, Proceedings of the 4th International Symposium on Visual Computing (ISVC08), pp. 170–180, Las Vegas, Nevada (USA), 2008

In this paper we first introduced our multi-dimensional Scale Saliency approach based on entropic graphs (see Chapter 4). The computational complexity of the original algorithm increases exponentially with data dimensionality, due to the fact that entropy estimation and self-dissimilarity computation steps are based on histograms. In our approach, Shannon's entropy is estimated from an approximation of the Rényi α-entropy when

$\alpha = 1$, by means of a MST spanning all the nodes representing the samples of the local pdf in the image. By the other hand, the Friedman and Rafsky test, an entropic graph estimation of the Henze-Penrose divergence, is used in order to compute self-dissimilarity between scales. Although complexity order decreases from exponential to linear with respect to data dimensionality, its execution time is still high.

Pablo Suau, Francisco Escolano, A New Feasible Approach to Multi-dimensional Scale Saliency, Proceedings of the 11th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS09), Bordeaux (France), 2009

This paper represents a new step towards devising a feasible multi-dimensional Scale Saliency algorithm. Not only the complexity order is linear with respect to data dimensionality, but also execution time is remarkably lower. The recent k-d partition algorithm is used during the entropy estimation step, remarkably increasing computational efficiency. Furthermore, a new divergence measure inspired by this algorithm is also presented.

Pablo Suau, Francisco Escolano, Entropy Estimation and Multi-dimensional Scale Saliency, to appear in the Proceedings of the 20th International Conference on Pattern Recognition (ICPR2010), Istanbul (Turkey), 2010

Pablo Suau, Francisco Escolano, Analysis of the Multi-dimensional Scale Saliency Algorithm and its Application to Texture Categorization, to appear in the Proceedings of the 8th International Workshop on Statistical Pattern Recognition (SPR2010), Istanbul (Turkey), 2010

In these papers we give a general survey of all the entropy and divergence estimation approaches used in our previous work (estimation from entropic graphs and the k-d partition algorithm). New experimental results validate our multi-dimensional Scale Saliency algorithm and our divergence measure

based on the total variant distance and the k-d partition method. An application of multi-dimensional Scale Saliency to texture categorization is also discussed.

A.2 Books

Francisco Escolano, Pablo Suau, Boyán Bonev, Information Theory in Computer Vision and Patter Recognition, Springer, 2009.

During year 2009 Springer published our book about the application of Information Theory to the fields of Computer Vision and Pattern Recognition. This book is not only a complete survey of the state of the art. It also reviews several of the contributions of our research group (including several contributions presented in this thesis) to the Information Theory community. Writing this book improved our knowledge about most of the measures, principles and theories related to Information Theory that were applied in this thesis.

Appendix B

Spanish version

B.1 Introducción

En este anexo presentamos un resumen de la tesis en español. Todos los contenidos de este anexo están explicados en mayor detalle en la versión en inglés.

B.1.1 Motivación

Dentro de la disciplina de la Inteligencia Artificial el objetivo del campo de la Visión Artificial es el diseño de sistemas o máquinas capaces de interpretar lo que *ven* mediante cualquier sensor que capte imágenes, como por ejemplo una cámara convencional. Se trata de un campo joven, siendo los primeros sistemas factibles de Visión Artificial diseñados durante los setenta. Debido a esto, el campo de la Visión Artificial es excitante: aparecen nuevas ideas constantemente, al igual que nuevas aplicaciones, como control de calidad industrial, análisis de imágenes médicas o localización y mapeado robótico. Algunas de estas aplicaciones incluso están empezando a dejarse ver en el mercado: cámaras con detección de caras, sistemas inteligentes de búsqueda y recuperación de imágenes, seguimiento en sistemas de videovigilancia, etc. Sin embargo, el campo de la Visión Artificial es todavía poco maduro. Los algoritmos de Visión Artificial tienden a resolver un tipo de problema muy específico, y la generalización de los mismos es muy difícil o imposible. En este sentido, la incorporación de métodos del Aprendizaje

Computacional puede proveer a los sistemas de Visión Artificial de nuevos métodos para la creación de sistemas adaptativos. Otra fuente importante de inspiración para los científicos del campo de la Visión Artificial es la visión biológica; la consecuencia de esto es la existencia de sistemas de Visión Artificial que tratan de imitar el sistema visual humano o el de los animales.

Las aplicaciones de Visión Artificial de alto nivel normalmente se apoyan en información de bajo nivel proporcionada por algoritmos de procesamiento de imagen o de extracción de características. Por ejemplo, las aristas de la imagen pueden convertirse en pistas esenciales durante la tarea de búsqueda de un determinado objeto en dicha imagen. Además, las características visuales extraídas proporcionan una representación dispersa de la imagen; las tareas de visión de alto nivel ya no tienen por qué procesar la imagen completa, sino que se pueden centrar en estas características, ahorrando tiempo de cómputo. Está claro por lo tanto que el rendimiento de los algoritmos de visión de alto nivel depende en gran medida del tipo y de la calidad de las características de bajo nivel extraídas. Esto significa que los algoritmos de extracción de características de bajo nivel son un elemento clave en la mayoría de sistemas de Visión Artificial. El objetivo de los algoritmos modernos de extracción de características es la búsqueda de características visuales que cumplan una serie de propiedades deseables, como ser informativas, distinguibles, e invariantes a transformaciones de la imagen, como cambios de escala, condiciones de iluminación o punto de vista. Algunos autores prefieren el término *covariantes*, refiriéndose a características visuales que se adaptan a la transformación aplicada a la imagen.

La Visión Artificial se relaciona con diversas disciplinas, como al Inteligencia Artificial, la Física, el Aprendizaje Computacional o las Matemáticas. La Teoría de la Información, una rama de las Matemáticas que estudia la cuantificación de la información, ha sido aplicada repetidas veces al campo de la Visión Artificial en el pasado. El algoritmo Scale Saliency es un ejemplo de algoritmo basado en Teoría de la Información. Se trata de un algoritmo de extracción de características que busca regiones salientes o altamente informativas en una imagen. Es un algoritmo interesante, no sólo porque experimentos previos han probado su buen rendimiento en el campo de la categorización de imágenes, sino que también porque tiene un buen

trasfondo teórico: aplica la Teoría de la Información con el objetivo de buscar las características de la imagen que son más informativas. Sin embargo, comparado con el resto de algoritmos de extracción de características invariantes dentro del estado del arte, se trata de un algoritmo bastante más lento. Esto restringe el uso del algoritmo a aplicaciones en las que no sea necesaria una respuesta inmediata.

B.1.2 Objetivos

El objetivo principal de esta tesis es mejorar la eficiencia del algoritmo Scale Saliency. El paso principal del algoritmo es la estimación de la entropía de cada píxel en el espacio de escalas definido por un determinado rango de escalas. Cada escala se representa como una región de tamaño creciente en el espacio de la imagen. El cuello de botella del algoritmo es precisamente la estimación de entropía en el rango completo de escalas. Nuestra hipótesis es que un subconjunto de píxeles de la imagen pueden ser marcados como no informativos, y por lo tanto descartados, incluso si la entropía ha sido estimada tan solo haciendo uso de una de las escalas. En este caso, el algoritmo Scale Saliency podría ser aplicado al resto de píxeles, ahorrando tiempo de cómputo.

Cabe preguntarse entonces qué escala debería ser procesada con tal de poder descartar puntos antes de aplicar el algoritmo Scale Saliency. Partimos de una idea intuitiva: si una región de la imagen es homogénea (y por lo tanto, no informativa), entonces subpartes de dicha región, a escalas más bajas, también serán homogéneas (y por lo tanto, no informativas). Por lo tanto, para poder descartar puntos antes de aplicar el algoritmo Scale Saliency deberíamos estudiar la máxima escala dentro del rango de escalas. Para apoyar esta idea necesitamos analizar como evoluciona la entropía a través del espacio de escalas. El objetivo de este análisis es encontrar una formalización para la relación entre la saliencia a altas y bajas escalas.

Una vez que dicha relación ha sido establecida, es necesario definir una regla de decisión que pueda ser usada para etiquetar píxeles como no interesantes, de tal forma que puedan ser filtrados antes de aplicar el algoritmo Scale Saliency. Esta regla de decisión debería ser general, o al menos aplicable a un conjunto de imágenes. Por lo tanto, lo que necesitamos

conseguir es un algoritmo de clasificación que permita, dado el valor de entropía de un píxel en la máxima escala, clasificarlo como útil o no para posteriores pasos del algoritmo Scale Saliency. Nuestro objetivo es aplicar la Teoría de la Información e inferencia Bayesiana para definir esta regla de clasificación, ya que ambos han sido anteriormente aplicados con éxito a problemas parecidos.

Otra meta de nuestra tesis, derivada de nuestro objetivo principal, es el estudio de la aplicación del algoritmo Scale Saliency a datos de alta dimensionalidad. Dicho algoritmo es aplicado comúnmente al procesamiento de imágenes en tonos de gris, pero debido a que la estimación de la entropía se realiza a partir de histogramas, es fácilmente aplicable a imágenes en color, e incluso a imágenes conteniendo información de mayor dimensionalidad. Sin embargo, la complejidad del algoritmo crece exponencialmente con la dimensionalidad de los datos, debido precisamente al uso de histogramas.

Es posible encontrar en la literatura métodos alternativos para la estimación de entropía, ya sea a partir de Árboles de Longitud Mínima, Grafos de K-Vecinos Más Cercanos u otros tipos de partición de datos. Necesitamos estudiar la idoneidad de este tipo de algoritmos para el paso de estimación de entropía del algoritmo Scale Saliency. Otro paso del algoritmo Scale Saliency que depende del uso de histogramas es la ponderación de picos de entropía. Como se explicará más adelante, el fin de este paso es penalizar aquellas características que sean salientes en un rango amplio de escalas. Dicho paso se basa en el uso de una medida de auto-disimilitud. Deberíamos por lo tanto evaluar diferentes métodos de estimación de divergencia a partir de grafos o mediante partición de datos, o incluso diseñar los nuestros propios, para evitar también el uso de histogramas durante este paso en el algoritmo Scale Saliency.

Un objetivo adicional (y también importante) de esta tesis es la aplicación de nuestras aportaciones a la resolución de problemas reales. Deberemos buscar aplicaciones para las que nuestras aportaciones al algoritmo Scale Saliency provean de claras mejoras en el rendimiento y/o la eficiencia.

B.1.3 Contenido de la tesis

Tras presentar las motivaciones y los objetivos de la tesis en la Sección B.1, presentamos en la Sección B.2 un estudio del estado del arte en el campo de los algoritmos de extracción de características en imágenes, centrándonos en algoritmos de extracción de puntos y regiones de interés. En este capítulo también definimos el concepto saliencia local y resumimos el funcionamiento del algoritmo Scale Saliency.

En la Sección B.3 presentamos una descripción detallada de nuestro algoritmo de filtrado Bayesiano, cuyo objetivo es la construcción de una regla de decisión que nos permita descartar puntos no interesantes antes de la aplicación del algoritmo Scale Saliency a una imagen. El algoritmo se basa en el análisis de la evolución de la entropía a lo largo del espacio de escalas. Nuestros experimentos muestran cómo afectan al funcionamiento de nuestro algoritmo el valor de determinados parámetros. También vemos cómo aplicarlo al problema de la localización robótica.

A continuación, en la Sección B.4, nos centramos en el diseño de una versión multidimensional del algoritmo Scale Saliency. Resumimos diferentes soluciones para la estimación de entropía y divergencia a partir de grafos y de partición de datos. Nuestros experimentos comparan nuestro algoritmo con el original en términos del número de características salientes detectadas, calidad de dichas características, y tiempo de ejecución. En este capítulo aplicamos nuestro Scale Saliency multidimensional al problema de la categorización de texturas.

En la Sección B.5 destacamos las principales conclusiones extraídas del trabajo presentado en esta tesis. También enumeramos diferentes aspectos de dicho trabajo que podrían estar sujetos a una futura investigación y mejora. Finalmente, en el Anexo A presentamos una lista de los artículos publicados en revistas y congresos a partir de los resultados presentados en esta tesis.

B.1.4 Aportaciones

A continuación enumeramos las principales aportaciones de la presente tesis:

- En primer lugar, en la Sección B.2, presentamos un estudio de la

evolución del estado del arte en el campo de los algoritmos de extracción de características en imágenes. A pesar de que podemos encontrar varios de estos estudios en la literatura, no estamos al corriente de ninguno de ellos que cubra la historia completa de estos algoritmos, desde los setenta hasta la actualidad. Además, resumimos algoritmos recientes que no han sido incluidos nunca en estudios anteriores.

- En segundo lugar, en la Sección B.3 presentamos un método cuyo objetivo es el incremento de la eficiencia temporal del algoritmo Scale Saliency. En primer lugar analizamos la evolución de la entropía a través del espacio de escalas. Este análisis es la base de un primer algoritmo de filtrado que permite descartar puntos de la imagen antes de aplicar el algoritmo Scale Saliency. Sin embargo, la aplicabilidad de este primer algoritmo es bastante limitada; no provee de ningún mecanismo para seleccionar a priori un umbral de entropía. Por ello la mejoramos por medio del uso de la Teoría de la Información y de inferencia Bayesiana, de tal forma que podamos aprender un umbral válido para un conjunto de imágenes. La Teoría de la Información también proporciona una herramienta de evaluación, la Información de Chernoff, que permite evaluar el funcionamiento de nuestro algoritmo de aprendizaje para el caso de un determinado conjunto de imágenes.

- En tercer lugar, en la Sección B.4 modificamos el algoritmo Scale Saliency de tal forma que conseguimos una disminución importante en su orden computacional (que pasa a ser de exponencial a lineal con respecto al número de dimensiones). Estudiamos diferentes métodos de estimación de entropía y divergencia cuya complejidad no depende en gran medida de la dimensionalidad de los datos, como sucede con en el algoritmo Scale Saliency debido al uso de histogramas. Estos métodos se basan en grafos y en partición de datos. Comparamos experimentalmente todos ellos para proponer un algoritmo Scale Saliency multidimensional factible, es decir, que permita procesar imágenes con información multidimensional en un tiempo razonable.

- Finalmente, aplicamos nuestros algoritmos a la resolución de

problemas reales. En la Sección B.3 vemos cómo aplicar nuestro algoritmo de filtrado al problema de la localización robótica. Con respecto al algoritmo de Scale Saliency multidimensional, en la Sección B.4 es aplicado al problema de la categorización de texturas.

B.2 Estado del arte en extracción de características en imágenes

Tal como se dijo en la introducción, el objetivo de esta tesis es buscar formas de mejorar la eficiencia temporal del algoritmo Scale Saliency. La función de este algoritmo de extracción de características es detectar las regiones salientes de una imagen. Estas regiones podrán más adelante ser usadas por sistemas de Visión Artificial de más alto nivel, como sistemas de categorización de imágenes o de localización robótica. En esta sección resumiremos el estado del arte en el campo de los algoritmos de extracción de características, e introduciremos algunos conceptos que serán utilizados en el resto de la tesis.

En primer lugar definiremos el concepto de punto de interés y resumiremos la historia de los algoritmos de extracción de este tipo de características. Mostraremos la evolución desde los sencillos algoritmos iniciales de Moravec y Förstner a los algoritmos actuales de extracción de características invariantes a cambios de escala y transformaciones afines, como el algoritmo Maximally Stable Extremal Regions o la versión afín del extractor de puntos esquina de Harris.

En segundo lugar definiremos el concepto de saliencia visual local, en el cual se basa el algoritmo Scale Saliency. La principal diferencia entre el algoritmo de Kadir y Brady y el resto de métodos presentados en nuestro resumen del estado del arte es que el primero se basa en la Teoría de la Información para modelar la saliencia o impredecibilidad. Veremos cómo este algoritmo surgió a partir de uno anterior creado por Gilles, cuyos resultados no eran invariantes a escala. La sección termina con una explicación detallada del algoritmo Scale Saliency.

B.2.1 Detección de características afines y multiescala en imágenes

En esta sección vamos a presentar la evolución de los algoritmos de extracción de características en imágenes, comenzando por los primeros métodos de este tipo, como el de Moravec [Moravec, 1977], el de Förstner [Förstner, 1986] o el de Harris [Harris and Stephens, 1988]. Veremos como se introdujo la invarianza a escala mediante representaciones del espacio de escalas como la de Witkin [Witkin, 1983]. Finalmente, veremos un pequeño resumen del estado del arte actual en extracción de características, donde se introduce el concepto de invarianza afín.

Puntos de interés

En las aplicaciones de Visión Artificial es habitual que en lugar de utilizar la imagen completa se extraiga un conjunto de características visuales de la misma. La extracción de características produce como resultado una representación dispersa de la imagen, de tal forma que se puede disminuir el tiempo necesario para procesarla. Por otra parte, estas características proporcionan información adicional. El tema de la extracción de características se encuentra presente en la literatura desde hace aproximadamente treinta años. La mayoría de los primeros algoritmos de este tipo estaban orientados a la detección de aristas en imágenes [Marr and Hildreth, 1980][Canny, 1986][Deriche, 1987]. El trabajo en esta tesis está basado en un algoritmo de extracción de características que extrae regiones de interés [Kadir and Brady, 2001]. Los algoritmos de extracción de regiones de interés provienen de trabajos previos de extracción de puntos de interés, que proporcionaron buenos resultados en los campos del emparejamiento estéreo o de la categorización de imágenes.

Un punto de interés es un píxel en la imagen cuyo nivel de información local es alto[1]. Estos puntos son distinguibles, y por lo tanto útiles en el contexto de los problemas de Visión Artificial. Además, los puntos de interés son diseñados para permanecer estables ante ciertos tipos de perturbaciones de la imagen, como variaciones en las condiciones lumínicas, cambio de punto de vista, etc. Esta estabilidad aumenta la robustez de las aplicaciones

[1]Otra definición de punto de interés es cualquier característica local para la que la señal varía en dos dimensiones [Schmid and Mohr, 1997]

de Visión Artificial que hacen uso de estas características. Pero también podemos encontrar otras ventajas de su uso. Por ejemplo, en el caso de detección de objetos, el uso de puntos de interés aumenta la robustez ante la oclusión de los objetos, ya que estos puntos de interés son extraídos a partir de partes locales de los mismos. Más que tratar de detectar un objeto como un todo, estos algoritmos permiten buscar pistas visuales de la presencia del objeto.

Podemos encontrar dos tipos de detectores de puntos de interés en la literatura: basados en geometría y basados en intensidad. En el primer caso el algoritmo normalmente se apoya en una extracción previa de algún tipo de característica geométrica, como aristas [Asada and Brady, 1986][Deriche and Faugeras, 1990]. En el segundo caso, los puntos de interés se extraen a partir de la intensidad de los píxeles de la imagen o a partir de gradientes de intensidad [Moravec, 1977][Rohr, 1990]. Resumiremos ahora algunos de estos algoritmos; para una revisión más completa se aconseja al lector consultar [Deriche and Giraudon, 1993].

Uno de los primeros algoritmos de detección de puntos esquina fue el **algoritmo de Moravec** [Moravec, 1977]. Dicho algoritmo se basa en calcular la diferencia de intensidades en tonos de gris entre los píxeles de una ventana 3×3 centrada alrededor del punto estudiado y los píxeles correspondientes en otras ventanas obtenidas al desplazar esta primera 1 píxel en las ocho direcciones principales. Se trata de un algoritmo rápido y sencillo, que según Harris y Stephens [Harris and Stephens, 1988] sufre de una serie de desventajas, entre ellas que se trata de un detector anisotrópico porque solo tiene en cuenta desplazamientos en ángulos de $45\,°$, o que las ventanas son binarias y cuadradas, con lo que la salida del algoritmo es más ruidosa que en el caso de haberse utilizado una ventana circular.

El **detector de Förstner** [Förstner, 1986] es aplicado a ventanas 5×5 ó 3×3 centradas en cada píxel de la imagen. El algoritmo se basa en el cálculo de una matriz de covarianza para cada ventana a partir de las intensidades en escala de grises de los píxeles en la misma. Esta matriz de covarianza representa una elipse. Para que el píxel en el que se encuentra centrada la ventana sea marcado como un punto de interés debe cumplir dos condiciones: su forma debe ser cercana a la de un círculo (lo cual se produce cuando los autovalores de la matriz de covarianza son similares) y el círculo debe ser

pequeño. Aunque es un extractor de características locales, el autor propone un método para buscar características globalmente significativas.

Sin duda uno de los extractores de puntos de interés más conocidos y utilizados es el conocido como **detector de Harris**, el cual fue presentado por Harris y Stephens en 1988 [Harris and Stephens, 1988]. Ha sido referenciado numerosas veces en la literatura, y aplicado a problemas como modelado 3D [Beardsley et al., 1996], categorización de imágenes [Leibe et al., 2004] o creación de mosaicos de imágenes [Capel and Zisserman, 1998]. Este algoritmo busca puntos esquina a partir de la matriz de autocorrelación ponderada con una ventana Gaussiana circular. La matriz de autocorrelación se obtiene a partir de las derivadas parciales de la imagen (gradientes) a lo largo de los ejes x e y. Los dos autovalores de esta matriz representan las dos variaciones principales de intensidad en la vecindad de un píxel. El tipo de característica a la que pertenece un píxel vendrá dada por el valor de estos autovalores. Si ambos son próximos a cero, el píxel no formará parte de ninguna característica reseñable de la imagen. Por otra parte, si el valor de uno de ellos es próximo a cero y el del otro es un número positivo alto, el píxel se hallará sobre una arista. Finalmente, si ambos autovalores son altos y positivos, se habrá encontrado un punto esquina. Para evitar el cálculo de los autovalores, que es costoso, Harris y Stephens proponen el uso de una función (normalmente denominada **función de Harris**) para medir el nivel de *esquinidad* de un píxel, a partir del determinante y la traza de la matriz de autocorrelación. Un ejemplo de aplicación del algoritmo se muestra en la Fig. 2.1.

En el caso del **detector de Noble** [Noble, 1988] los puntos esquina se definen como aquellos puntos en los que se produce una conjunción de aristas. La búsqueda se puntos esquina se basa en geometría diferencial y en la superficie topográfica que representa la función de intensidad de la imagen. Noble afirma que el detector de Harris es tan solo capaz de detectar esquinas en forma de L, mientras que su algoritmo permite además detectar esquinas en forma de X y T.

El **detector de Wang y Brady** [Wang and Brady, 1994] se basa en la búsqueda de extremos en la curvatura de la superficie de la imagen. La curvatura en un punto es proporcional a la segunda derivada de la imagen a lo largo de un vector $\mathbf{t}$ tangente a la superficie y perpendicular a su normal $\mathbf{n}$

en dicho punto. Ya que la curvatura se ve afectada por el ruido, se aplica un suavizado Gaussiano sobre la imagen, utilizando una baja desviación típica para evitar el desplazamiento de la localización de los puntos esquina detectados. Sólo se estudia la curvatura de aquellos puntos para los que la variación de la magnitud del gradiente, calculada mediante algún algoritmo de extracción de aristas como el de Sobel [Sobel and Feldman, 1968], es alto. De esta forma aumenta la eficiencia temporal del algoritmo y se evita la presencia de falsos positivos.

El **extractor de características de Lucas-Tomasi-Kanade** (ver [Tomasi and Kanade, 1991] y [Shi and Tomasi, 1994]) es un ejemplo de algoritmo de extracción de puntos de interés diseñado para una aplicación concreta; en este caso, el algoritmo busca las mejores características para la tarea de seguimiento (tracking) [Lucas and Kanade, 1981]. Según Tomasi *et al.*, el principal problema de las soluciones más generales es que se basan en una idea preconcebida del concepto de punto esquina o punto de interés. El análisis del problema de seguimiento llevó a Tomasi *et al.* a diseñar un algoritmo de extracción de características similar al detector de Harris, pero en el que la selección de puntos esquina se lleva a cabo mediante un test voraz.

El **algoritmo SUSAN** es un método rápido que se basa en un filtro no lineal y que al contrario que en el caso de las soluciones anteriores no requiere el uso de gradientes o núcleos Gaussianos. El algoritmo consiste en centrar un círculo de radio 3,4 (formado por 37 píxeles) alrededor de cada punto de la imagen. El punto en el centro del círculo recibe el nombre de núcleo, y el área dentro del círculo cuya intensidad sea similar a la del núcleo recibe el nombre de área USAN (Univalue Segment Assimilating Nucleus). Dependiendo del área USAN podremos clasificar el núcleo como un punto perteneciente a una arista (el área USAN ocupa la mitad del área del círculo) o un punto esquina (el área USAN ocupa menos de la mitad del área del círculo). La salida del algoritmo es un conjunto de puntos SUSAN (Smallest Univalue Segment Assimilating Nucleus).

El algoritmo **Minimum Intensity Change** [Trajkovic and Hedley, 1998] se basa en la variación de la intensidad a lo largo de un conjunto de líneas arbitrarias que pasan a lo largo del punto de estudio. Se usa la terminología del algoritmo SUSAN, comprobándose para cada línea arbitraria si sus

dos puntos de corte P y P' con el círculo centrado alrededor del núcleo pertenecen o no al área USAN. Si existe más de una línea para las que P y P' pertenecen al área USAN, el núcleo formará parte de un área homogénea de la imagen. Si existe tan solo una línea para la que esto suceda, el núcleo será parte de una arista. Finalmente, si para cualquier línea al menos uno de los puntos P o P' no pertenece al área USAN, tendremos un punto esquina.

Finalmente, destacamos el algoritmo **Features from Accelerated Segment Test** (FAST) [Rosten and Drummond, 2006] por ser reciente y estar basado en Aprendizaje Computacional y Teoría de la Información. El objetivo del algoritmo es diseñar un extractor de puntos esquina que pueda procesar imágenes en tiempo real. El resultado es un método que no sólo es más rápido que otros como el de Harris, SUSAN, y etc., sino que además proporciona puntos esquina de mayor calidad. El algoritmo FAST se basa en un filtro denominado *segment test*. Este filtro considera una circunferencia de 16 píxeles alrededor del candidato a punto esquina $\mathbf{p}$. Dicho punto candidato es etiquetado como punto esquina si al menos n píxeles contiguos de la circunferencia tienen una intensidad mayor o menor que $\mathbf{p}$. El algoritmo se basa en un proceso de aprendizaje en el que a partir de un conjunto de imágenes de entrenamiento se aprenden un conjunto de reglas que determinan qué píxeles de la circunferencia deben ir siendo estudiados, es decir, cuáles proporcionan mayor información acerca de la posibilidad de que $\mathbf{p}$ sea un punto esquina o no.

El espacio de escalas

Los detectores de puntos de interés resumidos arriba están basados en operadores que se aplican a regiones de tamaño fijo de la imagen. La consecuencia de esto es que tan solo detectan características pertenecientes a un rango pequeño de escalas. Sin embargo, en imágenes del mundo real estas características pueden ser encontradas a diferentes escalas. Además, disponer de la escala de las características detectadas puede aportar mayor información a los procesos de Visión Artificial que hagan uso de estas características. Por lo tanto, la principal desventaja de centrarse en la búsqueda de puntos de interés en una única escala es la pérdida de información importante sobre la imagen. La representación del espacio de

escalas surgió como un medio para detectar la presencia de estructuras locales a diferentes escalas.

La pirámide Gaussiana es una representación que se puede considerar la precursora del espacio de escalas. La pirámide permite procesar una imagen a nivel multiescala. El filtro definido por Burt [Burt, 1981] va aplicando de forma iterativa a la imagen un operador de suavizado Gaussiano y un submuestreo con el objetivo de obtener un conjunto de copias suavizadas de la imagen original a diferentes resoluciones. En cada nivel de la pirámide la resolución será menos fina, y por lo tanto el análisis se podrá centrar en estructuras de mayor tamaño.

El concepto de **espacio de escalas** fue introducido por primera vez por Witkin [Witkin, 1983], que diseñó un método para construir una descripción cualitativa de una señal 1D a diferentes escalas. Dada una señal 1D, y sin entrar a interpretar el significado de sus puntos extremos (que podrían indicar por ejemplo la presencia de aristas en una imagen), la escala puede ser modificada suavizando con un filtro Gaussiano. Sin embargo, el uso de este filtro produce ambigüedad ya que en cada nivel de escala los máximos presentes anteriormente pueden desaparecer, mientras que otros nuevos pueden aparecer. Además, el suavizado de la imagen puede producir un cambio en la localización de las estructuras locales de la imagen. Witkin afirma que las escalas no deben ser tratadas, por lo tanto, de manera individual, sino que como una representación global de la imagen. La representación de una imagen en el espacio de escalas produce un árbol que representa cualitativamente a la señal en cualquier escala de observación.

Witkin sugiere que su representación del espacio de escalas puede ser extendida al caso de una señal 2D. Fueron Yuille y Poggio [Yuille and Poggio, 1986] quienes finalmente lo hicieron, generalizando el concepto de espacio de escalas no solo al caso 2D sino que a cualquier número de dimensiones. Además, demostraron que para información 2D, la máscara Gaussiana 2D invariante a rotaciones es el único filtro de escalado que asegura la creación de nuevos extremos en el espacio de escalas sin la destrucción de ninguno de los presentes en escalas más bajas. Esto permite, una vez detectada una característica en una escala alta, realizar un seguimiento a través del espacio de escalas, hacia las escalas más bajas, para buscar su localización real. Estas mismas conclusiones fueron

extraídas por Koenderink y Richards [Koenderink and Richards, 1988], que afirmaron que construir el espacio de escalas de una imagen por medio de suavizados Gaussianos es equivalente a encontrar la solución a una función de difusión. Con esta representación, la distribución de intensidad de la imagen representa una distribución de temperaturas, de tal forma que la escala t representa la difusión a lo largo del plano de la imagen del calor en el instante t, asumiendo que la imagen es un material conductor.

El problema de todas estas representaciones es que no proveen de un método concreto para la selección de la escala característica de una determinada característica detectada en la imagen. Lindeberg propuso un marco *general* para la selección de la escala característica de una característica local de la imagen basado en la localización de máximos locales en el espacio de escalas y de la imagen simultáneamente. Este marco puede ser aplicado a diferentes tipos de características, como puntos esquina, aristas, blobs o intersecciones de aristas [Lindeberg, 1998][Lindeberg, 1994]. La principal aportación de este método es la posibilidad de detectar características **invariantes a cambios de escala**. Si se aplica un factor de escala s a una imagen, la escala característica de sus características locales también se multiplicará por el mismo factor s. Lindeberg afirma que la localización de una característica en las escalas más altas puede ser incorrecta, y propone una solución en dos fases. En la primera se detectan las características en las escalas más altas, y a continuación se refina su localización utilizando información de las escalas más bajas.

A pesar de que a continuación resumiremos algunos algoritmos de extracción de características con invarianza a cambios de escala, es interesante hacer notar que el espacio de escalas también ha sido aplicado para la creación de algoritmos de extracción de puntos esquina (sin invarianza a cambios de escala) más robustos. Un ejemplo es el **detector de Deriche y Giraudon** [Deriche and Giraudon, 1993], en el que el paradigma del espacio de escalas permite mejorar la localización de los puntos esquina detectados. Para ello, estudiaron un modelo de puntos esquina y su comportamiento en el espacio de escalas para diseñar un algoritmo que proporcionara la localización exacta de los mismos. Otro ejemplo es el **detector basado en ondas** de Loupias *et al.* [Loupias et al., 2000][Sebe and Lew, 2001]. Este algoritmo busca puntos de

interés por medio de transformadas de ondas, que son funciones oscilantes y atenuadas. El objetivo es que los puntos de interés detectados sean salientes en cualquier escala.

El primer método basado en la **Laplaciana de la Gaussiana** para detectar características multiescala fue obra de Blostein *et al.* [Blostein and Ahuja, 1989]. Dicho método se basa en la búsqueda de máximos de la función Laplaciana en el espacio de escalas. Una vez un máximo es encontrado, se acepta como característica saliente si la vecindad del punto se puede ajustar a un disco cuyo diámetro sea similar al del núcleo Gaussiano utilizado para detectar dicho punto en la escala correspondiente.

Un ejemplo de método de detección de puntos esquina adaptado al paradigma del espacio de escalas es el algoritmo **Harris-Laplace** de Mikolajczyk y Schmid [Mikolajczyk and Schmid, 2001]. Se basa en el modelo de espacio de escalas de Witkin [Witkin, 1983]. Se aplica exactamente la misma matriz de correlación que en el caso de la detección de puntos esquina de Harris, solo que a cada escala de manera individual. La escala característica de una determinada estructura local de la imagen se obtiene como el máximo sobre el espacio de escalas de la función de *esquinidad* para dicha estructura. El problema es que para una misma estructura local la función de *esquinidad* produce máximos locales en diferentes escalas **y** localizaciones. La solución es aplicar un algoritmo iterativo para refinar la posición y la escala de las estructuras detectadas a partir de un conjunto inicial de las mismas. En este paso algunas de las características convergen. Un ejemplo de aplicación del algoritmo se muestra en la Fig. 2.3.

El algoritmo SIFT (Scale Invariant Feature Transform) de Lowe [Lowe, 1999][Lowe, 2004] es ampliamente utilizado como método para la descripción de características detectadas en imágenes, de tal forma que diferentes características puedan ser comparadas entre sí. Sin embargo, este algoritmo fue concebido en un principio como un algoritmo completo de detección y descripción de características. La fase de detección de SIFT se basa en el cálculo de **diferencias de Gaussianas**. En primer lugar se crea una representación del espacio de escalas de la imagen mediante suavizados Gaussianos. A continuación, se calcula la diferencia entre la salida del suavizado Gaussiano para cada par de imágenes en niveles consecutivos del espacio de escalas. La diferencia de Gaussianas es eficiente

computacionalmente y además proporciona una aproximación de la Laplaciana de una Gaussiana normalizada con respecto a la escala. Los candidatos a ser detectados como características salientes en la imagen serán aquellos que se correspondan a un máximo local en la función de diferencia de Gaussianas. Estos candidatos serán posteriormente refinados para encontrar el conjunto final de características salientes de la imagen.

Un método más actual es el algoritmo **Robust Invariant Features** de Lin *et al.* [Lin et al., 2005]. Primero se buscan extremos de la función de *esquinidad* de Harris a lo largo del espacio de escalas. Como la localización de una característica de la imagen varía con la escala, se estudia la evolución de la localización de estas características en el espacio de escalas, de tal forma que se pueda hallar una localización más exacta. A la evolución de una característica a lo largo del espacio de escalas se le denomina *Local Corner Signature* (LCS). Durante el estudio del espacio de escalas, varios máximos pueden llegar a formar parte de un mismo LCS, por lo que se debe determinar si se corresponden o no con la misma estructura local dentro de la imagen.

Invarianza afín

El principal inconveniente de los métodos multiescala es que, debido al hecho de que se usan núcleos Gaussianos isotrópicos para construir la representación del espacio de escalas, el ángulo entre el vector de la vista y el vector normal a la superficie sobre la que se encuentra la característica detectada puede ser estimado erróneamente. Este factor es crítico en el caso de algunas transformaciones, como por ejemplo el cambio de punto de vista, que producen un gran cambio en este ángulo. Las regiones isotrópicas (circulares) no pueden representar este tipo de transformaciones. Es necesario el uso de regiones anisotrópicas (elipses).

Como en el caso de la invarianza a escala, Mikolajczyk y Schmid también adaptaron el detector de Harris para que fuera invariante a transformaciones afines [Mikolajczyk and Schmid, 2002][Mikolajczyk and Schmid, 2004b], creando lo que se conoce como el **detector Harris afín**. Debido a la gran carga computacional que supondría construir un espacio de escalas no uniforme, el algoritmo busca las características afines en la vecindad de las características previamente detectadas mediante la versión multiescala

del algoritmo, por medio de un método iterativo basado en el trabajo de Lindeberg y Gårding [Lindeberg and Gårding, 1997]. Este método consiste en ir construyendo de manera iterativa una matriz de transformación, concatenando varias matrices de transformación diferentes basadas en la matriz de covarianza de Harris, de tal forma que la región detectada, una vez transformada, cumpla una serie de condiciones.

Un método muy parecido al anterior es el seguido por el **detector Hessiano afín** [Mikolajczyk and Schmid, 2004a]. La única diferencia con el Harris afín es que en lugar de basarse en una matriz de covarianza para la estimación de la *esquinidad* de los diferentes píxeles de la imagen, hace uso de una matriz Hessiana, construida a partir de las segundas derivadas de la imagen suavizada en ambos ejes. Se puede ver un ejemplo de aplicación de estos dos algoritmos en la Fig. 2.4.

Un algoritmo extractor de características que ha recibido mucha atención desde su creación es el conocido como **Maximally Stable Extremal Regions** (MSER), de Matas *et al.* [Matas et al., 2004]. Una MSER es una componente conexa dentro de una imagen umbralizada, para la que la intensidad de todos los píxeles que la forman es mayor o menor que la de todos los píxeles en su contorno. Se trata de un método simple y poco costoso computacionalmente. En primer lugar, se ordenan los píxeles de la imagen en orden ascendente o descendente de intensidad. A continuación, estos píxeles van siendo incluidos en una estructura de tipo *union-find* que almacena una lista de componentes conexas y su intensidad; el área de las componentes conexas se establece en función de la intensidad. El paso final es buscar para cada componente conexa los niveles de intensidad que producen mínimos locales en el ritmo de cambio de crecimiento del área. Las características así detectadas normalmente tienen formas arbitrarias, aunque suelen ser transformadas a elipses. En la Fig. 2.5 se puede observar un ejemplo de aplicación del algoritmo.

El algoritmo **Maximally Stable Corner Clusters** (MSCC), de Fraundorfer *et al.* [Fraundorfer et al., 2005], está fuertemente inspirado en el anterior. La diferencia principal con respecto a los extractores de características citados hasta ahora es que cada región es construida a partir de un conjunto de puntos de interés. El primer paso consiste en la extracción de estos puntos de interés. Fraundorfer calcula para cada píxel de la imagen su *esquinidad*

de Harris y su tensión estructural [Bigun and Granlund, 1987], intentando que se seleccione una gran cantidad de puntos. Cada uno de estos puntos de interés se corresponderá con un nodo de un Árbol de Longitud Mínima ponderado, siendo el peso de cada arista la distancia entre sus dos vértices incidentes. A continuación, inspirándose en el algoritmo MSER, el algoritmo va incrementando un umbral de peso, de tal forma que se eliminan todas aquellas aristas cuyo peso sea superior. Esto produce una serie de subárboles. Cada uno de estos subárboles se corresponde con una región de la imagen. Las regiones salientes serán aquellas cuyo tamaño permanezca estable a lo largo de varios incrementos del umbral de peso. Como en el caso del algoritmo MSER, las características detectadas tendrán forma arbitraria; podrán ser transformadas en elipses si se desea. Se puede ver un ejemplo de aplicación en la Fig. 2.6.

Tuytelaars *et al.* definieron dos métodos basados en la detección de puntos semilla a partir de los cuales se buscan las regiones con invarianza afín [Tuytelaars and Gool, 2004]. Estos métodos son complementarios, en el sentido en el que las características detectadas por un algoritmo no suelen ser detectadas por el otro y viceversa. El primero de ellos, **Edge Based Regions** (EBR), está basado en puntos esquina de Harris y aristas extraídas con el detector de Canny [Canny, 1986]. Sea un punto esquina de Harris $\mathbf{p}$ detectado sobre una arista, y sean dos puntos $\mathbf{p}_1$ y $\mathbf{p}_2$ que se van moviendo en direcciones opuestas por dicha arista, a la misma velocidad. Los tres puntos forman un paralelogramo. Los puntos $\mathbf{p}_1$ y $\mathbf{p}_2$ se detienen, y una característica es detectada, cuando alguna propiedad fotométrica del paralelogramo alcanza un valor extremo (estas propiedades podrían ser por ejemplo funciones basadas en momentos invariantes). En su segundo algoritmo, **Intensity Based Regions** (IBR), se estudia la intensidad a lo largo de un conjunto de rayos rectos que emanan desde cada punto semilla, siendo los puntos semilla máximos locales de la intensidad de la imagen. Se busca el máximo a lo largo de cada rayo, enlazándose todos ellos para formar una región afín alrededor del punto semilla. Las regiones extraídas mediante EBR e IBR también pueden ser transformadas a elipses. La Fig. 2.7 muestra un ejemplo de aplicación de ambos algoritmos.

B.2.2 Extracción de características basada en saliencia visual local

En esta sección definiremos brevemente el concepto de saliencia visual local y presentaremos el algoritmo Scale Saliency de Kadir y Brady [Kadir and Brady, 2001]. Todas las aportaciones de nuestra tesis se basan en estos dos elementos.

La palabra saliencia hace referencia a la cualidad de ser saliente. Una entidad determinada es saliente si destaca sobre el resto de entidades pertenecientes al mismo dominio. Esta definición puede ser directamente trasladada al contexto del procesamiento de imágenes. Diremos que una región en una imagen será visualmente saliente si es claramente distinguible del resto de elementos de la imagen, ya sea en términos de intensidad, orientación, o cualquier otra propiedad. Podemos ver dos ejemplos de este concepto en la Fig. 2.8. En la parte izquierda de la imagen se puede observar un círculo verde que destaca sobre el resto de elementos de la imagen, por lo que será saliente con respecto al resto de círculos. En la parte de la derecha, el elemento saliente es aquel que tiene una orientación diferente al resto. Ambos elementos podrían ser considerados salientes tanto globalmente (porque destacan con respecto al resto de elementos de la imagen) como localmente (porque destacan con respecto a los elementos situados en su vecindad).

Con el objetivo de extraer características útiles de las imágenes, nuestros algoritmos estarán centrados en la detección de regiones salientes con respecto a la intensidad local, es decir, regiones cuya distribución de intensidad sea claramente diferente a la de las regiones en su vecindad. Es de suponer que la mayoría de estas regiones seguirán siendo salientes aunque apliquemos diferentes transformaciones a la imagen, como por ejemplo cambio de escala o de condiciones de iluminación [Kadir et al., 2004]. Esta propiedad es imprescindible si deseamos construir aplicaciones de Visión Artificial que se apoyen en el uso de estas características.

El algoritmo de Gilles

Gilles fue el primer autor en relacionar la saliencia visual local en la extracción de características en imágenes con la Teoría de la información [Gilles, 1998]. Él propuso utilizar la entropía de Shannon para estimar saliencia en imágenes. La entropía de Shannon es una medida de

la impredecibilidad asociada a una variable aleatoria. Básicamente hace referencia a la cantidad de información contenida en un mensaje. Cuanto menos predecible sea el valor de una variable aleatoria, más información proporcionará. Gilles definió el concepto de regiones salientes en imágenes de tonos de gris como regiones impredecibles con respecto a su contexto local, es decir, aquellas regiones que proveen mayor información sobre la imagen (regiones de alta entropía). Dado un pixel $\mathbf{x}$, cuya intensidad toma un valor en el dominio $D = \{d_1, \ldots, d_n\}$ (valores en el rango entre 0 y 255 en el caso de una imagen en tonos de gris), y su vecindad local $R_{\mathbf{x}}$, su saliencia puede ser estimada por medio de la entropía de Shannon [Cover and Thomas, 1991]:

$$H_{D,R_{\mathbf{x}}} = -\sum_i P_{D,R_{\mathbf{x}}}(d_i) \log_2 P_{D,R_{\mathbf{x}}}(d_i) \ , \tag{B.1}$$

donde $P_{D,R_{\mathbf{x}}}$ es la proporción de píxeles en $R_{\mathbf{x}}$ cuya intensidad es d_i. Los valores bajos de entropía se corresponden con variables aleatorias con un bajo nivel de información, es decir, variables aleatorias en las que la probabilidad de un valor determinado es mucho mayor que la del resto de valores. Por otra parte, los valores altos de entropía se corresponden con variables aleatorias impredecibles, para las que la probabilidad de todos sus posibles valores es similar. Si tratamos el problema de extracción de características desde un punto de vista basado en Teoría de la Información, es obvio que lo que deben hacer este tipo de algoritmos es buscar regiones de la imagen con un valor alto de entropía. La Fig. 2.9 muestra un ejemplo de como esta métrica caracteriza las regiones salientes y no salientes. En esta figura se extrae un histograma de frecuencias de tonos de gris para dos regiones diferentes, calculándose la entropía a partir de los mismos. Como se puede observar, la predecibilidad en las regiones homogéneas de la imagen es alta (el histograma de intensidades tiene un pico marcado), y por lo tanto su saliencia es baja. Este hecho es básico para entender las aportaciones que presentamos en la Sección B.3.

El algoritmo de extracción de características de Gilles funciona de la siguiente manera: en primer lugar se establece un tamaño y una forma fijos para $R_{\mathbf{x}}$. En nuestros ejemplos utilizamos regiones cuadradas de radio fijo, medido en píxels. Este radio es lo que se conoce como escala. Se estima la entropia para cada píxel utilizando la Ec. B.1. Todos aquellos píxeles para los

que su valor de entropía esté por debajo de un umbral son descartados. Esto produce como resultado un conjunto de componentes conexas. El algoritmo finalmente selecciona el máximo local en cada uno de estas componentes conexas. Estos puntos se corresponderán con las características salientes de la imagen. En la Fig. 2.10 mostramos un ejemplo de aplicación del algoritmo, usando una escala de 7 y un umbral de entropía de 6,6. La mayoría de las características salientes en la figura se corresponden con coches o partes de coches, ya que éstos destacan localmente sobre la carretera, que es más homogénea.

Aunque la estimación de saliencia basada en entropía es intuitiva y sencilla, el algoritmo de Gilles sufre algunas limitaciones [Kadir and Brady, 2001]. La más evidente es el uso de una escala fija. Debido a esto, la búsqueda de elementos salientes en la imagen está restringida a un pequeño rango de escalas. Se puede ver un ejemplo claro de este problema en la imagen de los coches de la Fig. 2.10. El peatón de la parte superior derecha de la imagen es saliente con respecto al pavimento; sin embargo, la escala de esta característica de la imagen es demasiado pequeña y no se considera como una característica saliente por el algoritmo. Otro problema del algoritmo de Gilles es que es muy sensible al ruido. Finalmente, las regiones altamente texturizadas de la imagen, con grandes cambios de intensidad, podrían seleccionarse como características salientes, aunque se diera el caso de que dichas regiones formaran parte de una región texturizada más grande y no fueran por lo tanto salientes desde un punto de vista visual.

El algoritmo Scale Saliency

Kadir y Brady propusieron su algoritmo Scale Saliency con la idea de superar las limitaciones del algoritmo de Gilles. Su nueva propuesta buscaba regiones salientes no solo en el espacio de la imagen sino que también en el espacio de escalas. Para analizar el espacio de escalas se estima la entropía para cada píxel, siguiendo el método de Gilles, mientras se va aumentando el tamaño de su vecindad de manera isotrópica. Por lo tanto, la salida del algoritmo Scale Saliency no es un conjunto de puntos salientes (como en el caso del método de Gilles), sino que un conjunto de círculos o regiones de

diferentes tamaños. A pesar de que Kadir y Brady propusieron también una versión anisotrópica del algoritmo [Kadir et al., 2003], a lo largo de esta tesis nos centramos en la versión isotrópica del mismo. Debido a que el algoritmo Scale Saliency analiza el espacio de escalas, se hace necesaria una redefinición del término saliencia visual, que se puede encontrar sujeta a dos posibles interpretaciones: una región podría ser considerada saliente tanto si lo es en un rango amplio de escalas como si lo es en un rango pequeño de escalas. El algoritmo de Kadir y Brady adopta la segunda definición, ya que según los autores el hecho de que se produzca saliencia en un rango amplio de escalas es una consecuencia de estar analizando una región de la imagen correspondiente a una estructura fractal, o con una intensidad totalmente aleatoria, o a la presencia de una alta auto-similitud. Para ser considerada saliente una región debe ser distinguible tanto en el espacio de la imagen como en el espacio de escalas.

El algoritmo funciona de la siguiente forma. En primer lugar, se establece un rango de escalas entre una escala mínima s_{min} y una escala máxima s_{max}. A continuación se estima la entropía de cada píxel $\mathbf{x}$ en cada escala s a partir de su función de densidad de probabilidad de intensidad:

$$H_D(s, \mathbf{x}) = -\sum_{d \in D} P_{d,s,\mathbf{x}} \log_2 P_{d,s,\mathbf{x}} \ . \tag{B.2}$$

En la Fig. 2.11 se muestra un ejemplo de estimación de entropía a diferentes escalas para una figura sintética compuesta por círculos, usando un rango de escalas definido entre $s_{min} = 5$ y $s_{max} = 20$. En dicha figura las tonalidades más claras representan valores más altos de entropía.

Una vez estimada la entropía para cada píxel, se seleccionan regiones salientes candidatas. Estas regiones candidatas se corresponderán con aquellas escalas de un determinado pixel para las que se alcanza un máximo local, o un pico de entropía:

$$S_p = \{s : H_D(s-1, \mathbf{x}) < H_D(s, \mathbf{x}) > H_D(s+1, \mathbf{x})\} \ . \tag{B.3}$$

La entropía de un punto $\mathbf{x}$ en las escalas $s \in S_p$ es ponderada por medio de una métrica de auto-similitud en el espacio de escalas de la característica. Esta métrica de auto-similitud permite una comparación directa de la

saliencia de diferentes píxeles a diferentes escalas, además de penalizar aquellas características que son salientes en un rango amplio de escalas:

$$W_D(s, \mathbf{x}) = \frac{s^2}{2s-1} \sum_{d \in D} |P_{d,s,\mathbf{x}} - P_{d,s-1,\mathbf{x}}| \ . \tag{B.4}$$

En la Fig. 2.12 se muestra un ejemplo de cálculo de entropía ponderada a diferentes escalas para aquellos puntos y escalas detectados como picos de entropía, en el caso de la imagen sintética de la Fig. 2.11. En la Fig. 2.13 se puede ver también cómo esta métrica de auto-similitud permite establecer correctamente la escala de las regiones salientes (en la parte de la izquierda se muestra el resultado del algoritmo Scale Saliency al no aplicar la ponderación de la Ec. B.4, mientras que en la parte de la derecha se muestran los resultados cuando sí se aplica dicha ponderación).

Una vez explicados estos pasos, podemos definir la saliencia de una región como su entropía ponderada. Esta saliencia, como se ha comentado anteriormente, se calcula para todas las escalas seleccionadas s_p de cada píxel $\mathbf{x}$:

$$Y_D(s_p, \mathbf{x}) = H_D(s_p, \mathbf{x}) W_D(s_p, \mathbf{x}) \ . \tag{B.5}$$

Por lo tanto, la salida del algoritmo Scale Saliency es una matriz $Y(S, X)$ que almacena la saliencia de cada píxel en las escalas seleccionadas. Las características más salientes de la imagen se corresponderán con los valores máximos en $Y(S, X)$. En la Fig. 2.15 podemos ver ejemplos de aplicación del algoritmo a diferentes imágenes, incluida la imagen de los coches de la Fig. 2.10. El algoritmo Scale Saliency detecta ahora correctamente el peatón y el cubo de basura situado a su lado como características salientes, aunque su escala sea menor que la de los coches.

Un proceso adicional muy útil es el que se denomina supresión de no máximos, o agrupamiento de características. El objetivo del agrupamiento de características es el de unir la información de las características salientes que se encuentren próximas en el espacio de la imagen. Este paso reduce el número de características detectadas, lo que hace descender la complejidad espacial y temporal de las aplicaciones que utilicen este tipo de características. También hace aumentar la robustez del algoritmo [Kadir and Brady, 2001]. El agrupamiento de características se

desarrolla siguiendo los pasos que se enumeran a continuación, teniendo en cuenta que K y V_{th} son parámetros cuyos valores han sido preestablecidos:

1. Escoger la región más saliente de $Y(S, X)$
2. Encontrar los k vecinos más cercanos
3. Calcular la varianza V del centro de todas estas regiones y la escala media s_{mean}, así como la posición media $\mathbf{x}_{mean}$.
4. Encontrar la distancia D en $\mathcal{R}^3$ (fila, columna y escala) desde la región seleccionada a las regiones que ya han sido agrupadas.
5. Crear una nueva región saliente si $D > s_{mean}$ y si $V < V_{th}$. Esta región saliente se añade al conjunto de regiones ya agrupadas, siendo su localización $\mathbf{x}_{mean}$ y su escala s_{mean}.
6. Repetir a partir del segundo paso con la siguiente región más saliente, hasta que una determinada cantidad de los elementos de $Y(S, X)$ haya sido procesada.

En la Fig. 2.14 se puede ver un ejemplo de aplicación del algoritmo Scale Saliency, incluyendo el paso de agrupamiento de características, a la imagen en la Fig. 2.1. Nótese que en este caso las regiones extraídas son isotrópicas, y no regiones afines.

B.3 Filtrado Bayesiano en el algoritmo Scale Saliency

B.3.1 Introducción

En esta sección nos centramos en la mayor desventaja del algoritmo Scale Saliency cuando lo comparamos con el resto de algoritmos de extracción de características en el estado del arte: su baja eficiencia temporal [Mikolajczyk et al., 2005b]. Incluso en el caso en el que se reutilicen cálculos entre escalas para construir los histogramas que servirán para estimar las funciones de densidad de probabilidad de intensidad (ver Sección B.3.6), el cuello de botella del algoritmo sigue siendo la estimación de la entropía de Shannon (Ec. B.1) para todos los píxeles en todas las

escalas dentro del rango $[s_{min}, s_{max}]$. El tiempo de ejecución del algoritmo de Kadir y Brady es notablemente más alto que el del resto de extractores de características.

Este hecho motivó nuestro estudio de la evolución de la entropía en el espacio de la imagen y a lo largo del espacio de escalas, con el objetivo de demostrar una hipótesis sencilla e intuitiva: aquellas regiones de la imagen que sean homogéneas o no salientes en las escalas más altas seguramente también lo serán en el resto del espacio de escalas. La idea de esta sección es aprovechar las propiedades de la entropía de tal forma que podamos descartar puntos no interesantes, es decir, puntos que probablemente no serán parte de las características más salientes de una imagen, antes de aplicar el algoritmo Scale Saliency al resto de la misma. En esta sección la Teoría de la Información nos proporciona una serie de medidas para evaluar la aplicabilidad de nuestro algoritmo de filtrado a un conjunto de imágenes y para estimar la probabilidad de error, es decir, la probabilidad de que una característica saliente sea descartada, dependiendo del valor de un determinado parámetro.

En primer lugar, en la Sección B.3.2 mostramos los resultados de nuestro estudio de la evolución de la entropía en el espacio de la imagen y a lo largo del espacio de escalas. También demostramos nuestra hipótesis mediante inferencia estadística, y como consecuencia de ello, presentamos una primera aproximación al filtrado de puntos de la imagen de manera previa a la aplicación del algoritmo Scale Saliency en la Sección B.3.3. Esta primera aproximación no tiene utilidad práctica: el valor de su único parámetro (un umbral de entropía) puede ser tan solo calculado *a posteriori*, una vez que el algoritmo Scale Saliency ha sido aplicado a la imagen completa; para una imagen diferente necesitaríamos un valor distinto. A partir de esta primera aproximación al problema, y basándonos en trabajos anteriores de detección estadística de aristas y contornos [Konishi et al., 2003][Cazorla et al., 2002], presentamos en la Sección B.3.4 los fundamentos de nuestro algoritmo de filtrado, que se resume en la sección B.3.5. Gracias a nuestro algoritmo de filtrado Bayesiano, podemos utilizar un conjunto de imágenes pertenecientes a una misma categoría o entorno para obtener un umbral *a priori*, de tal forma que podamos usarlo para filtrar nuevas imágenes. En nuestro algoritmo la Teoría de la Información es utilizada para validar el agrupamiento de

imágenes y para estimar la probabilidad de error dado un valor de umbral. En la Sección B.3.6 mostramos varios resultados experimentales. Finalmente, en la Sección B.3.7 mostramos una aplicación práctica de nuestro algoritmo en el campo de la localización robótica.

B.3.2 Análisis de la entropía en el espacio de escalas

En esta sección resumiremos las conclusiones que nos llevaron la hipótesis que desarrollamos a lo largo de toda la sección B.3: *regiones homogéneas o no salientes en la escala máxima probablemente también serán homogéneas o no salientes en escalas más bajas*. Estas conclusiones fueron extraídas a partir de la observación empírica de la evolución de la entropía en el espacio de escalas y de la imagen y a partir de inferencia estadística. En la Fig. 3.1 se puede ver, por ejemplo, una representación tridimensional de la saliencia de la imagen de los coches (ver Fig. 2.10) usando cuatro escalas diferentes. Como se puede ver, la saliencia no cambia bruscamente entre píxeles adyacentes; al contrario, su evolución en el espacio de escalas es suave, mucho más en el caso de las escalas más altas.

Este hecho nos motivó a comprobar la suavidad de la evolución de la entropía en el espacio de escalas. El objetivo de esta comprobación era obtener la suficiente evidencia empírica de tal forma que pudiéramos comenzar un estudio más profundo que confirmara nuestra hipótesis. Para ello representamos la evolución de la entropía a lo largo del espacio de escalas en el caso de diferentes imágenes de entrada. Un ejemplo se muestra en la Fig. 3.2, en la que se puede observar la evolución de la entropia a lo largo del espacio de escalas para todos los píxeles situados en dos filas diferentes de la imagen de los coches. Esto se hace así porque para representar la evolución para la imagen completa necesitaríamos una gráfica en cuatro dimensiones. Es por ello que en dicha figura nos centramos en dos filas de la imagen, mostrando como varía la entropía entre las escalas $s_{min} = 5$ y $s_{max} = 15$. Como se puede ver en la figura, el valor de la entropía cambia suavemente a lo largo del espacio de escalas para cualquier pixel. La consecuencia de ello es que aquellos píxeles para los cuales el valor de la entropía es bajo en s_{max} también tienen valores bajos de entropía en s_{min}. También demostramos empíricamente esto en la Fig. 3.3, en la que las regiones salientes en la

escala 5 en la imagen de los coches (izquierda) se sobreimponen sobre las regiones salientes en la escala 20 de la misma (derecha). Como se puede ver, la correspondencia es alta. Todas estas observaciones fueron también realizadas tomando otras imágenes de entrada.

Tras estas pruebas preliminares, realizamos una serie de experimentos destinados a probar la conexión entre las características más salientes de la imagen y sus valores de entropía en las escalas s_{min} y s_{max}. Concretamente, nuestra hipótesis era la siguiente:

- Si la entropía de un pixel en s_{min} **y** en s_{max} es alta, seguramente formara parte de las características más salientes de la imagen.
- Si la entropía de un pixel en s_{min} **o** en s_{max} es alta, también es probable que sea parte de las características más salientes de la imagen, pero con una probabilidad menor que en el caso anterior.
- Aquellos píxeles cuya entropía tanto en s_{min} como en s_{max} sea baja probablemente no formarán parte de las regiones más salientes de la imagen.

Para nuestros experimentos utilizamos el conjunto de imágenes *Object categories*[2], publicado por el *Visual Geometry Group* de la Universidad de Oxford. Este conjunto de imágenes está formado por 12 categorías, conteniendo cada una de 126 a 1155 imágenes de diferentes tamaños. Para nuestro experimento se escogieron aleatoriamente 240 imágenes (20 por categoría), y de cada una de ellas se seleccionó, también aleatoriamente, 1000 píxeles (haciendo un total de 240000 píxeles). A continuación, usando el algoritmo de Kadir y Brady (ver Sección B.2.2), estimamos la entropía de todos estos puntos en el rango de escalas entre $s_{min} = 5$ y $s_{max} = 20$. Para probar nuestra hipótesis definimos una serie de variables:

- f_3: ratio entre H_{min} y h_{min}
- f_5: ratio entre H_{max} y h_{max}
- f_7: ratio entre H_{min} y h^*

[2]http://www.robots.ox.ac.uk/~vgg/

- f_9: ratio entre H_{max} y h^*

donde, para un determinado píxel, h_{min} es su entropía en s_{min}, h_{max} es su entropía en s_{max}, y h^* es su máximo valor de entropía en el espacio de escalas, y donde H_{min} es la máxima entropía de la imagen en s_{min} y H_{max} es la máxima entropía de la imagen en s_{max}. La correlación estándar entre estas variables se muestra en la Tabla 3.1. Centramos nuestro análisis en las relaciones mostradas en negrita en dicha tabla. Como se puede observar, la correlación entre las variables estudiadas es fuerte (con un valor próximo a 1) en todos los casos. Esto indica una fuerte dependencia lineal entre cada par de variables, y sugiere que la entropía de un punto en s_{min} y s_{max} puede ayudar a localizar puntos pertenecientes a las características más salientes. Por otra parte, los valores de correlación para f_5 son mayores que los de f_3, lo que parece indicar que h_{max} proporciona más información acerca de la probabilidad de que un punto sea parte de las regiones más salientes que h_{min}.

La Fig. 3.5 muestra gráficamente la relación entre los pares de variables cuyo valor de correlación aparece en negrita en la Tabla 3.1, basándonos en los valores de entropía de los puntos seleccionados del conjunto de imágenes del Visual Geometry Group. Lo más interesante de la figura es que los valores de f_3 y f_5 parecen proporcionar un límite inferior de la máxima entropia de un pixel a lo largo del espacio de escalas. Esto significa que la probabilidad de que un píxel sea parte de las características más salientes de la imagen es más alta en el caso de que dicho píxel tenga un valor alto de saliencia en s_{max} y s_{min}. Sin embargo, la regresión lineal (la línea roja en la Fig. 3.5) no proporciona información útil.

Otra medida que apoya la idea de una fuerte dependencia entre las variables en nuestro estudio es la correlación múltiple. El nivel de dependencia de una variable y con respecto a dos variables independientes x_1 y x_2 viene dado por

$$R = \sqrt{\frac{r_{y,x_1}^2 + r_{y,x_2}^2 - 2r_{y,x_1}r_{y,x_2}r_{x_1,x_2}}{1 - r_{x_1,x_2}^2}} \ , \tag{B.6}$$

donde $r_{a,b}$ es el valor de correlación entre las variables a y b. En nuestro caso, si tomamos f_3 y f_5 como variables independientes, el valor de la correlación

múltiple con f_7 y f_9 es 0,9926 y 0,9964, respectivamente. Este resultado sugiere que este tipo de relación entre las variables es todavía más fuerte que en el caso de la dependencia lineal. La Fig. 3.6 muestra una representación tridimensional de esta relación. Tanto f_3 como f_5 proporcionan un límite inferior similar al de la Fig. 3.5.

El límite inferior en la parte inferior de la Fig. 3.6 es prácticamente un plano. La ecuación de dicho plano se correspondería con el mínimo valor esperado de la saliencia máxima de un píxel a lo largo del espacio de escalas. Dicha ecuación podría ayudar a definir un test formal que permitiera descartar puntos de la imagen que probablemente no son parte de las regiones más salientes. La consecuencia de esto es que el algoritmo Scale Saliency no necesitaría explorar todo el espacio de escalas, y por lo tanto su complejidad temporal descendería. Para estimar los parámetros del plano aplicamos una transformada de Hough [Hough, 1962][Duda and Hart, 1972][Vosselman and Dijkman, 2001] de tal forma que pudiéramos aproximar el límite inferior. Dada la ecuación del plano en $\mathcal{R}^3$

$$f_9 = s_x f_3 + s_y f_5 + d \ , \tag{B.7}$$

los parámetros que se deben estimar son la pendiente del plano en el eje x (s_x), la pendiente del plano en el eje y (s_y) y la altura del plano en el eje z con respecto al origen de coordenadas (d). El algoritmo de la transformada de Hough se basa en una matriz tridimensional de votos, en la cual cada celda se asocia a un subconjunto discreto del espacio $\{s_x, s_y, d\}$. Cada punto del plano correspondiente al límite inferior que queremos aproximar añade un voto en todas las celdas asociadas a todos los posibles planos que pasan por dicho punto. Para disminuir los requisitos temporales y espaciales del algoritmo restringimos el espacio de búsqueda a aquellos valores cercanos al plano esperado. Finalmente, la celda con más votos nos dará los parámetros del plano. El plano obtenido se muestra en la Fig. 3.7.

Los parámetros que se obtuvieron para el plano fueron $s_x = 0$, $s_y = 1.01$ y $d = 0.015$. La conclusión es que mientras que el valor de h^* depende en gran medida de los valores de h_{min} y h_{max}, el límite inferior de h^* tan solo parece depender de h_{max}. Trasladando esta conclusión al caso del algoritmo

Scale Saliency, podemos decir que existe una fuerte dependencia entre la probabilidad de que un píxel forme parte de las regiones más salientes de la imagen y su valor de entropía en s_{max}. Esto demuestra nuestra idea intuitiva de que las regiones homogéneas de la imagen en las escalas más altas también lo serán en las escalas más bajas. Es muy poco probable que estas regiones homogéneas contengan ninguna característica saliente.

B.3.3 Una primera solución de filtrado

En la Sección B.3.2 hemos demostrado que la entropía de un punto en la escala s_{max}, dado un rango de escalas $[s_{min}, s_{max}]$, es suficiente para estimar la probabilidad de que un píxel sea parte de las características más salientes de la imagen. Dado H_{max} (la máxima entropía entre todos los píxeles de la imagen en s_{max}), todos aquellos píxeles cuya entropía h_{max} en s_{max} sea cercana a H_{max} ($h_{max}/H_{max} \approx 1$) probablemente serán parte de las regiones más salientes de la imagen. En esta sección presentamos una primera solución simple para disminuir la carga computacional del algoritmo Scale Saliency, basada en estos resultados. Consiste en establecer un valor para un umbral σ, de tal forma que todos los píxeles para los que $h_{max}/H_{max} < \sigma$ serán descartados de forma previa a la aplicación del algoritmo de Kadir y Brady. El establecer un umbral de entropía normalizado con respecto a H_{max} nos va a ser útil para escoger un único umbral σ que pueda ser aplicado para filtrar varias imágenes diferentes (ver la Sección B.3.4). Los pasos del algoritmo de filtrado son:

1. Calcular la entropía local H_D en la escala s_{max} para todos los píxeles $\mathbf{x}$ de la imagen
2. Seleccionar un umbral de entropía normalizada $\sigma \in [0, 1]$
3. $X = \left\{ \mathbf{x} : \frac{H_D(\mathbf{x}, s_{max})}{\max_{\mathbf{x}}\{H_D(\mathbf{x}, s_{max})\}} > \sigma \right\}$
4. Aplicar el algoritmo Scale Saliency sólo a $\mathbf{x} \in X$

La principal dificultad en este algoritmo es establecer un umbral σ adecuado para cada imagen. Un valor muy alto hará que se descarten algunos píxeles que sí formen parte de las características más salientes de

la imagen. Por otra parte, un umbral muy bajo podría producir el filtrado de una baja cantidad de píxeles, y como consecuencia, se podría producir un incremento en el tiempo de ejecución con respecto al algoritmo Scale Saliency original. En la Fig. 3.8 mostramos ejemplos de aplicación del algoritmo de filtrado a la imagen de los coches. En la parte izquierda se ven los resultados del algoritmo Scale Saliency, en la parte central el resultado de aplicar un umbral de $\sigma = 0.73$ (los puntos que fueron filtrados se muestran en rojo) y en la parte derecha el resultado de aplicar un umbral de $\sigma = 0.82$. En todos los casos el rango de escalas se estableció entre $s_{min} = 5$ y $s_{max} = 20$ y se muestran las 200 características más salientes (tras aplicar el proceso de supresión de no máximos). Conforme aumenta σ, más puntos son descartados y menor es el tiempo total de ejecución. Si σ es demasiado alto (como en el caso de la parte derecha de la Fig. 3.8, el algoritmo Scale Saliency produce falsos negativos y positivos.

La Fig. 3.9 muestra el efecto del aumento del umbral σ en el número de puntos filtrados y en el tiempo de ejecución comparado con el algoritmo Scale Saliency original. En la parte de la izquierda se muestran los resultados para la imagen de los coches, y en la parte de la derecha se muestran los resultados para la imagen sintética de la Fig. 2.11. En ambos casos, la línea discontinua vertical muestra el mayor umbral que es posible utilizar sin que se produzcan falsos positivos o negativos, en el caso de seleccionar las 200 características más salientes. La imagen sintética se compone de un mayor número de regiones homogéneas; por lo tanto, con un umbral bajo ya se consigue un notable descenso en el tiempo de ejecución. Debido a la cantidad de información contenida en imágenes del mundo real, el máximo umbral posible que no produce errores en el caso de la imagen de los coches es mayor y consigue filtrar una menor cantidad de puntos. Además, el descenso acelerado en el tiempo de ejecución se produce también para umbrales más altos.

Para determinar el máximo umbral libre de errores en el experimento anterior partimos de un análisis estadístico similar al del trabajo de detección de aristas de konishi *et al.* [Konishi et al., 2003]. Este análisis implica la definición de dos distribuciones denominadas $P(\theta|on)$ y $P(\theta|off)$. La primera representa la probabilidad de que un píxel forme parte de las características más salientes de la imagen dado que su valor de

entropía normalizado en s_{max} sea θ. Por otra parte, $P(\theta|off)$ representa la probabilidad de que un píxel no sea parte de las características más salientes de la imagen dado que su entropía normalizada en s_{max} sea θ. En la Fig. 3.10 mostramos ambas distribuciones en el caso de las imágenes de la Fig. 3.9 (en la parte izquierda las correspondientes a la imagen de los coches y en la parte derecha las correspondientes a la imagen sintética). Estas distribuciones se estimaron **tras** aplicar el algoritmo Scale Saliency a todos los puntos de la imagen. El máximo umbral posible libre de errores viene dado por el primer valor de θ, comenzando por $\theta = 0$, para el que $P(\theta|on) \neq 0$. Dado que es necesario aplicar el algoritmo Scale Saliency antes de estimar el umbral, este método no es el más apropiado para realizar dicha estimación. Además, el umbral más adecuado puede ser diferente para imágenes distintas. La Fig. 3.10 también proporciona pruebas adicionales a lo dicho en la Sección B.3.2: los píxeles con mayor entropía en s_{max} es más probable que formen parte de las regiones más salientes de la imagen. Esto es más cierto todavía en imágenes del mundo real.

B.3.4 Filtrado Bayesiano e Información de Chernoff

En esta sección introducimos el marco teórico sobre el que se construye nuestro algoritmo de filtrado. El objetivo es explotar la información estadística de las imágenes para aprender un umbral que sea válido para un conjunto de imágenes diferentes. Este método se apoya en la idea de que imágenes pertenecientes a un mismo entorno o una misma categoría de objetos comparten propiedades como intensidades, texturas y estadísticos de segundo orden [Torralba and Oliva, 2003][Farinella et al., 2008]. Por lo tanto, el valor de la entropía de sus características más salientes también se encontrará aproximadamente dentro del mismo rango. Dado un conjunto de imágenes pertenecientes a un mismo entorno o categoría, y siguiendo el método estadístico de Konishi *et al.* [Konishi et al., 2003] (que más tarde fue extendido por Cazorla *et al.* [Cazorla et al., 2002][Cazorla and Escolano, 2003]), aprendemos un umbral para dicho conjunto. Para ello estimamos, a partir de una selección de imágenes de entrenamiento, el par de distribuciones $P(\theta|on)$ y $P(\theta|off)$ para dichas imágenes, tal como hicimos en la Sección B.3.3 para imágenes

individuales (ver la Fig. 3.10). Este análisis estadístico también nos será útil para estudiar el riesgo asociado al proceso de filtrado.

Nuestro método es aplicable tan solo en el caso de un conjunto homogéneo de imágenes. Esto nos lleva a preguntarnos: ¿cómo podemos comprobar si un conjunto de imágenes es lo suficientemente homogéneo como para poder extraer información estadística útil a partir del mismo? En esta tesis proponemos aplicar la medida conocida como Información de Chernoff [Cover and Thomas, 1991]. La Información de Chernoff $C(P,Q)$ entre dos distribuciones P y Q se define como

$$C(P,Q) = -\min_{0\leq\lambda\leq 1} \log\left(\sum_{j=1}^{J} P^{\lambda}(y_j)Q^{1-\lambda}(y_j)\right) \quad , \tag{B.8}$$

donde $\{y_j : j = 1,\ldots,J\}$ son las variables aleatorias sobre las que se definen las distribuciones. La Información de Chernoff mide lo fácil que es discriminar entre dos distribuciones de probabilidad. En la Fig. 3.11 mostramos las distribuciones $P(\theta|on)$ y $P(\theta|off)$ para dos categorías de imágenes del conjunto de imágenes del Visual Geometry Group (en la parte izquierda para la categoría *airplanes_side* y en la parte derecha para la categoría *camel*). Estas distribuciones se estimaron a partir de un conjunto de imágenes de entrenamiento aleatoriamente seleccionado para cada categoría. En el caso de la categoría *airplanes_side* se obtuvo $C(P(\theta|on),P(\theta|off)) = 0.4$, mientras que en el caso de la categoría *camel* se obtuvo $C(P(\theta|on),P(\theta|off)) = 0.14$. El valor de la Información de Chernoff es alto cuando $P(\theta|on)$ y $P(\theta|off)$ son claramente separables, es decir, cuando dada la entropía normalizada de un píxel en s_{max} es sencillo determinar si dicho píxel formará parte de las características más salientes de la imagen. Si el valor de la Información de Chernoff es bajo, entenderemos que el conjunto de imágenes no es suficientemente homogéneo y que debería ser dividido en diferentes subconjuntos antes de aplicar nuestro método de filtrado. En esta tesis no tratamos el problema de la partición óptima de las imágenes en categorías. Este es un tema que ha sido incluido en la sección de trabajo futuro (ver Sección B.5.2).

Ahora tratamos el problema de cómo escoger un umbral adecuado para una categoría de imágenes dada sus distribuciones $P(\theta|on)$ y $P(\theta|off)$. Este

tipo de análisis fue previamente aplicado a la detección y el agrupamiento de aristas [Konishi et al., 2003][Cazorla and Escolano, 2003]. En este caso, $P(\theta|on)$ y $P(\theta|off)$ representaban la probabilidad de que un píxel formara parte o no de una arista dependiendo de la respuesta de un determinado filtro θ. Basándose en el trabajo de detección de carreteras de Geman y Jedynak [Geman and Jdeynak, 1996], Konishi *et al.* afirman que la similitud logarítmica de estas dos distribuciones puede ser utilizada como una medida de fuerza de arista, es decir, como una medida de probabilidad de existencia de una arista [Konishi et al., 2003]. A partir de este razonamiento, aplican una prueba de similitud logarítmica

$$\log \frac{P(\theta|on)}{P(\theta|off)} > T \ , \tag{B.9}$$

siendo T un umbral, para determinar si un píxel forma parte de una arista dada la respuesta de un determinado filtro θ. En esta tesis proponemos aplicar este mismo test para realizar un filtrado de píxeles de la imagen previo a la aplicación del algoritmo Scale Saliency. En nuestro caso, θ representa la entropía normalizada del píxel en s_{max}. Si escogemos un umbral $T \geq 0$, estaremos descartando todos aquellos píxeles para los que $P(\theta|on) < P(\theta|off)$. Por otra parte, si $T < 0$, algunos puntos cuya probabilidad de no formar parte de las regiones más salientes de la imagen sea baja no serán filtrados. Una vez más, es necesario decidirse por un umbral para el filtrado, lo cual puede llevarnos a pensar que tenemos el mismo problema que en el caso de la primera solución de filtrado presentada en la Sección B.3.3. Sin embargo, la ventaja de este test es que podremos establecer un rango de valores válidos de T para un determinado conjunto de imágenes.

Tal como se indica en [Cazorla and Escolano, 2003], sabemos que el umbral T debe cumplir

$$-D(P(\theta|off)||P(\theta|on)) < T < D(P(\theta|on)||P(\theta|off)) \ , \tag{B.10}$$

donde $D(p||q)$ es la divergencia de Kullback-Leibler entre las distribuciones p y q:

$$D(p||q) = \sum_{j=1}^{J} p(y_j) \log \left(\frac{p(y_j)}{q(y_j)} \right) \ . \tag{B.11}$$

La divergencia de Kullback-Leibler, o entropía relativa, mide la disimilitud entre dos distribuciones. Se podría interpretar como una estimación de la pérdida de eficiencia de codificación en el caso de asumir que la distribución subyacente es q cuando realmente es p [Cover and Thomas, 1991]. En estadística se define también como el valor esperado del logaritmo del ratio de similitud.

Los diferentes valores del umbral en el rango definido por la Ec. B.10 producen diferentes resultados durante el filtrado. Si se selecciona el mínimo valor dentro del rango se obtendrá un filtro conservador que asegurará un buen equilibrio entre una baja tasa de error (un número bajo de falsos positivos y negativos) y una alta eficiencia temporal. Si se selecciona un valor de T más alto, tanto la eficiencia temporal como el error aumentarán. Las medidas de la Teoría de la Información presentadas en esta sección nos permitirán estimar cómo afectan al error la información estadística de una categoría de imágenes. El primer hecho a tener en cuenta es que la probabilidad de error aumenta exponencialmente con $C(P(\theta|on), P(\theta|off))$ [Cover and Thomas, 1991]. Además, la Información de Chernoff y la divergencia de Kullback-Leibler están relacionadas. Como se indicó anteriormente, si el valor de $C(P(\theta|on), P(\theta|off))$ es bajo, las dos distribuciones serán similares y será más difícil establecer un umbral válido. La consecuencia es que el rango de posibles valores de T definido por la Ec. B.10 será más estrecho; por lo tanto, menos puntos podrán ser filtrados, y el error será mayor.

Una ventaja adicional del test de similitud logarítmica es que, en general, sus resultados son mejores. En la Fig. 3.12 comparamos los resultados de filtrado del test de similitud logarítmica con los del test sencillo presentado en la Sección B.3.3, aplicados a una imagen de la categoría *bottles* (en la parte izquierda se ve la salida del algoritmo Scale Saliency, en la parte central el resultado del filtrado aplicando el test de la Sección B.3.3 con $\sigma = 0.7$, con el que se consiguieron filtrar un 23% de los píxeles de la imagen, y en la parte derecha el resultado del filtrado aplicando en test de similitud logarítmica con $T = -0.65$, con lo que se filtraron un 26% de píxeles de la imagen). En el caso del filtrado basado en el test de similitud logarítmica, aprendimos un rango de umbrales *general* para toda la categoría *bottles* (ver Sección B.3.5) y aplicamos el máximo umbral posible que no produjera ningún falso positivo

o negativo. En el caso del filtrado simple se utilizó el máximo umbral, *obtenido exclusivamente para dicha imagen*, que no produjo ningún falso positivo o negativo.

B.3.5 Filtrado Bayesiano previo al Scale Saliency: el algoritmo

En esta sección resumimos nuestro algoritmo de filtrado, el cual está basado en los conceptos explicados en secciones anteriores. Suponiendo que las imágenes de entrada están divididas en clases o categorías, el primer paso consiste en el aprendizaje de un umbral para cada una de ellas. Una vez hecho esto, los umbrales aprendidos pueden ser usados para descartar píxeles de imágenes pertenecientes a esas mismas clases antes de aplicar el algoritmo Scale Saliency, consiguiéndose una notable reducción del tiempo de ejecución y una baja tasa de error (ver Sección B.3.6).

A continuación resumimos los pasos necesarios para aprender un umbral a partir de un conjunto de imágenes pertenecientes a una misma clase o categoría:

1. Calcular las distribuciones de probabilidad $P(\theta|on)$ y $P(\theta|off)$ a partir de todos los píxeles del conjunto de imágenes de entrenamiento, determinando si dichos píxeles pertenecen o no a las características más salientes de sus correspondientes imágenes, y siendo

$$\theta = \frac{H_D(\mathbf{x}, s_{max})}{\max_{\mathbf{x}}\{H_D(\mathbf{x}, s_{max})\}} \tag{B.12}$$

   el valor de la entropía normalizada del píxel $\mathbf{x}$ en la escala s_{max}.

2. Evaluar la Información de Chernoff entre estas dos distribuciones. Si el valor de $C(P(\theta|on), P(\theta|off))$ es demasiado bajo, la categoría no es lo suficientemente homogénea y no es posible aprender un buen umbral para ella. En ese caso, dividir la clase de imágenes en dos o más subclases y repetir el proceso de entrenamiento para cada una de ellas.

3. Calcular las divergencias de Kullback-Leibler $D(P(\theta|off)||P(\theta|on))$ y $D(P(\theta|on)||P(\theta|off))$.

4. Seleccionar un umbral en el rango $-D(P(\theta|off)||P(\theta|on)) < T < D(P(\theta|on)||P(\theta|off))$. El valor mínimo de T en ese rango es un umbral conservativo que proporciona un buen equilibrio entre una alta eficiencia temporal y una baja tasa de error. Los valores más altos de T harán aumentar el error dependiendo del valor de $C(P(\theta|on), P(\theta|off))$ [Konishi et al., 2003].

Una vez realizado este paso previo de aprendizaje, nuevas imágenes de la categoría podrán ser filtradas antes de aplicarles el algoritmo Scale Saliency, de tal forma que se descartarán píxeles que probablemente no formarán parte de las características más salientes de la imagen:

1. Calcular la entropía normalizada local $\theta_{\mathbf{x}}$ en la escala s_{max} para cada píxel $\mathbf{x}$ (ver Ec. B.12).

2. Seleccionar el conjunto de puntos de interés

$$X = \left\{ \mathbf{x} | \log \frac{P(\theta_{\mathbf{x}}|on)}{P(\theta_{\mathbf{x}}|off)} > T \right\} \quad , \tag{B.13}$$

   donde T es el umbral aprendido para la categoría a la que pertenece la imagen de entrada.

3. Aplicar el algoritmo de Kadir y Brady tan solo a los píxeles $\mathbf{x} \in X$.

En la Sección B.3.6 mostraremos algunos ejemplos de aplicación. Realizaremos también diferentes experimentos destinados a demostrar la validez de nuestra solución. Por último comprobaremos el efecto de los parámetros del algoritmo en el resultado final.

B.3.6 Resultados experimentales

Con el fin de demostrar la validez de nuestra solución de filtrado realizamos una serie de experimentos utilizando el conjunto de imágenes *Object Category* del Visual Geometry Group (para una descripción de este conjunto de imágenes, ver la Sección B.3.2). En nuestros experimentos respetamos la partición original en categorías de las imágenes, aunque más tarde se demostró que la Información de Chernoff era baja para alguna de ellas, es decir, que existen categorías que no son suficientemente homogéneas.

En estos casos la eficiencia del algoritmo podría aumentar dividiendo esas categorías en subcategorías.

Para cada categoría estimamos sus distribuciones $P(\theta|on)$ y $P(\theta|off)$, usando un 10% de imágenes de cada una de ellas como conjunto de entrenamiento. El rango de escalas se estableció entre $s_{min} = 5$ y $s_{max} = 20$. Las imágenes fueron escaladas antes de la aplicación del algoritmo: ni la altura ni la anchura de cualquier imagen podría ser superior a 320 píxeles. Con respecto al número de elementos de los histogramas, se utilizaron 128 en lugar de los 256 correspondientes a los 256 diferentes niveles de intensidad de una imagen en tonos de gris. La cuantización de los histogramas reduce notablemente el tiempo de ejecución, y no afecta en gran medida a los resultados finales. Kadir y Brady también aplican esta técnica en su implementación, en la que utilizan tan solo 64 elementos. Por último, se estableció que el número de características salientes a extraer fuera de 50, aunque en la mayoría de los casos el número de características finalmente obtenido es menor, debido al proceso de supresión de no máximos o agrupamiento de características descrito en la Sección B.2.2.

La Tabla 3.2 muestra los resultados del experimento de filtrado usando el conjunto de imágenes previamente mencionado. Dicha tabla muestra el porcentaje medio de puntos filtrados y de tiempo ahorrado por imagen para dos umbrales diferentes: el mínimo posible dentro del rango de umbrales válidos (Ec. B.10) y $T = 0$. La última columna muestra el error medio por imagen, calculado de la siguiente forma:

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}\frac{d(A_i, B_i) + d(B_i, A_i)}{2} \quad , \tag{B.14}$$

donde n es el número de imágenes de la categoría, $A = \{A_1, \ldots, A_n\}$ es el conjunto de regiones más salientes obtenidas por el algoritmo Scale Saliency para cada imagen de la categoría (tras la supresión de no máximos), $B = \{B_1, \ldots, B_n\}$ es el conjunto de características más salientes obtenidas con nuestro método para cada imagen en la categoría (tras la supresión de no máximos), y donde

$$d(A, B) = \sum_{a \in A} \min_{b \in B} ||a - b|| \tag{B.15}$$

es una medida de la distancia entre las características en A y en B basada en la distancia Euclídea. La distancia en la Ec. B.15 se calcula en $\mathcal{R}^3$: esto quiere decir que se considera el error tanto en el espacio de la imagen como en el espacio de escalas.

En general se obtienen mejores resultados para aquellas categorías con un alto valor de Información de Chernoff: o bien se descartan más puntos o bien el error es menor. Por ejemplo, los valores más altos de Información de Chernoff fueron obtenidos para las categorías *airplanes_side*, *cars_brad_bg* y *leaves*. Usando el mínimo umbral T válido para cada una de ellas se puede ahorrar hasta un 35 – 40% de tiempo de proceso. Los valores más bajos de Información de Chernoff se obtuvieron para las categorías *bottles*, *camel* y *google_things*. En todos estos casos tan solo se ahorra un 20 – 25% de tiempo, aunque esto sigue siendo un ahorro significativo. Es extremadamente raro encontrar alguna imagen para la que el proceso de filtrado seguido del algoritmo Scale Saliency requiera mayor tiempo de ejecución que el algoritmo Scale Saliency original. Por otra parte, el umbral $T = 0$ proporciona unos resultados todavía mejores en cuanto a tiempo ahorrado para todas las categorías, sin que el error aumente en gran medida. El valor de T dependerá de la aplicación. El mínimo valor válido de una categoría proporciona un buen equilibrio entre una alta velocidad de ejecución y un bajo error. El valor $T = 0$ podría ser utilizado en aplicaciones como localización robótica, en las que la eficiencia computacional es un factor clave [Newman et al., 2006].

En la Fig. 3.13 mostramos ejemplos de aplicación de nuestro método a tres imágenes pertenecientes a tres categorías diferentes. En la columna de la izquierda se muestra la salida del algoritmo Scale Saliency, en la parte central la salida de nuestro método utilizando el mínimo umbral válido de la categoría a la que pertenece la imagen, y en la parte derecha la salida de nuestro método utilizando $T = 0$. En general se suele descartar el fondo de la imagen si éste es homogéneo. También se pueden descartar regiones texturizadas o no homogéneas, debido al uso de entropía normalizada y al hecho de que se aprende un umbral diferente para cada categoría. Además, nuestro algoritmo no necesariamente filtra más puntos en el caso de categorías con una alta presencia de regiones homogéneas. Por ejemplo, casi todas las imágenes dentro de la categoría *bottles* tienen un fondo uniforme, pero los resultados para esta categoría son peores que los obtenidos para la

categoría *houses*, en la que la presencia de texturas es mayor.

También aplicamos nuestro método a las distintas categorías de la base de datos de imágenes *Caltech101*[3], que consta de 101 categorías, conteniendo cada una de ellas entre 40 y 800 imágenes. Para cada categoría se seleccionaron aleatoriamente 20 imágenes de entrenamiento y 20 de test. Se aprendió un rango de umbrales validos para cada categoría a partir de sus imágenes de entrenamiento. Finalmente, calculamos el cantidad media de puntos filtrados y el error medio por imagen para cada categoría usando dos umbrales: el mínimo umbral válido para cada categoría y $T = 0$. El experimento se repitió 10 veces. Los resultados se muestran en la Fig. 3.15, y también son resumidos en la Tabla 3.3, donde destacamos las categorías para las que nuestro algoritmo proporciona los mejores y los peores resultados.

En general no hay ahorro de tiempo si la cantidad de puntos filtrados de la imagen es inferior al 20%. De hecho, en esos casos el algoritmo completo, incluyendo el filtrado, requiere mayor tiempo de computación que el algoritmo Scale Saliency original. Sin embargo, esto sucede en menos del 40% de las categorías de la base de datos de imágenes y cuando T es el mínimo posible para cada categoría. Cuando $T = 0$ nuestro método siempre produce una reducción en el tiempo de ejecución. Con respecto al error, aquellas categorías para las que la cantidad media de puntos filtrados es mayor no necesariamente se corresponden con las categorías en las que el error es mayor. Hay excepciones, como en el caso de la categoría *stapler*,pero hemos de tener en cuenta que en ese caso se filtra de media un 82,10% de puntos de cada imagen. Como se podría esperar, los valores de error más bajos no se corresponden con los valores de filtrado más altos, excepto en algunos casos como en el de la categoría *Faces*, que es la séptima para la que más puntos se filtran, teniendo un valor de error de 0 (para el caso del mínimo umbral válido).

El resto de experimentos que se realizaron tuvieron como objetivo determinar el efecto de determinados parámetros en el resultado final del algoritmo. Por ejemplo, se estableció que se debía utilizar un 10% de imágenes de cada categoría del Visual Geometry Group para el entrenamiento después de llevar a cabo un experimento en el que se

[3]http://www.vision.caltech.edu/Image_Datasets/Caltech101

calculó la Información de Chernoff de cada categoría dependiendo de la cantidad de imágenes de entrenamiento. La Información de Chernoff ha sido aplicada anteriormente como medida de la calidad de un clasificador [Konishi et al., 2003]. Los resultados del experimento se pueden ver en la Fig. 3.16. El valor de la Información de Chernoff disminuye rápidamente conforme aumentamos la proporción de imágenes de cada categoría utilizadas como imágenes de entrenamiento de un 2% a un 10%. Para proporciones mayores, el valor de la Información de Chernoff no varía o lo hace muy suavemente. Es por ello que escogimos un 10% de imágenes de entrenamiento: en el caso de proporciones menores no se tiene suficiente información para obtener un rango válido de umbrales, ya sea porque el valor de la Información de Chernoff evoluciona muy rápidamente o de manera aleatoria. Por otra parte, proporciones mayores no proporcionan información adicional al proceso de aprendizaje, y por lo tanto no se produce ninguna mejora.

Otro parámetro a tener en cuenta es el número de características salientes a extraer por el algoritmo, ya que el rango de umbrales válidos para una categoría depende de las distribuciones $P(\theta|on)$ y $P(\theta|off)$, que a su vez dependen de este parámetro. Si su valor aumenta, el solapamiento entre $P(\theta|on)$ y $P(\theta|off)$ será mayor, disminuyendo el valor de la Información de Chernoff y por lo tanto haciendo que el rango de umbrales válidos sea más estrecho. Esto se puede ver en la Fig 3.17, en la que se muestran las distribuciones $P(\theta|on)$ y $P(\theta|off)$ de la imagen de los coches en el caso en el que el número de características a detectar sea 50 (parte izquierda, $C(P(\theta|off), P(\theta|on)) = 0,57$) y en el caso en el que el número de características a detectar sea 400 (parte derecha, $C(P(\theta|off), P(\theta|on)) = 0,29$). En la Tabla 3.4 y en la Fig. 3.18 mostramos el efecto de cambiar el número de características a detectar para el caso de la categoría *bottles*. Cuando este número se encuentra entre 50 y 300, el valor de la información de Chernoff va disminuyendo lentamente, volviendo a aumentar de nuevo para un número mayor de características. Esto quiere decir que si se seleccionan menos características será más fácil discriminar qué regiones de la imagen serán salientes o no, con lo que el número de píxeles filtrados será mayor. En el caso en el que se decida extraer un alto número de características, un mayor número de píxeles con un valor alto de entropía relativa en s_{max} pasan a ser

consideradas en la estimación de $P(\theta|on)$, con lo que su solapamiento con $P(\theta|off)$ empieza a disminuir. Por los resultados también se puede concluir que este parámetro no afecta en gran medida en los resultados. Esto se ve en la Fig. 3.18, donde aplicamos nuestro método a una imagen de la categoría *bottles* con el mínimo umbral posible para el caso de diferente número de características a mostrar.

El rango de escalas es un parámetro importante en el algoritmo Scale Saliency, pues afecta directamente a su tiempo de ejecución. Este parámetro debería ser establecido después de estudiar en qué rango de escalas suelen encontrarse las características para una categoría dada. Por ejemplo, en la Fig. 3.20 mostramos la frecuencia de escalas de las características salientes para todas las categorías del conjunto de imágenes *Object category*, cuando $s_{min} = 5$ y $s_{max} = 20$. Como se puede observar, las características más salientes se concentran en el rango definido entre $s_{min} = 5$ y $s_{max} = 15$. Por lo tanto podríamos haber utilizado dicho rango de escalas para obtener los resultados mostrados en la Tabla 3.2. El resultado se muestra en la Tabla 3.5. En todos los casos la cantidad media de puntos filtrados aumentó entre dos y cinco veces. Con respecto al error, éste no necesariamente incrementó, e incluso en el caso de algunas categorías fue menor. Este experimento demuestra que un análisis previo del espacio de escalas para el conjunto de imágenes para el que se va a aplicar el algoritmo Scale Saliency puede mejorar los resultados del filtrado.

B.3.7 Ejemplo de aplicación: localización robótica

Quizá la limitación más destacable de nuestro algoritmo sea que es necesario conocer la categoría o el entorno al que pertenece una imagen antes de poder aplicarle el filtrado. Esta limitación podría evitar que aplicáramos nuestro algoritmo a problemas reales, como por ejemplo la categorización de imágenes. Sin embargo, nuestro algoritmo puede ser útil en otros contextos. En esta sección vemos cómo pudimos aplicar con éxito nuestra versión del algoritmo Scale Saliency al problema de la localización robótica. Los resultados de esta sección se obtuvieron gracias a la colaboración con otros investigadores dentro de ese campo [Escolano et al., 2007].

El problema de la localización robótica (ver [Arkin and Balch, 1998]

o [Thrun et al., 2001]) se puede definir de la siguiente manera: en primer lugar, disponemos de un robot que almacena una base de datos de imágenes tomadas del entorno por medio de una cámara situada sobre el mismo. Una vez construida la base de datos, se deja al robot navegar por dicho entorno. Durante la navegación del robot, y con el objetivo de conocer su localización, éste toma una nueva imagen y busca aquella de su base de datos que sea más similar. Esta última imagen es la salida del algoritmo. Hay que tener en cuenta que la capacidad de localizarse en el entorno ha sido considerada una de las más importantes para poder considerar a un robot como totalmente autónomo [Cox, 1989]. En nuestros experimentos utilizamos el mapa de la Fig. 3.21, que representa una reconstrucción 3D de algunas áreas del edificio de la Politécnica III y sus alrededores en la Universidad de Alicante. Dicho mapa se obtuvo mediante un algoritmo de mapeado y localización simultáneos con 6 grados de libertad [Sáez and Escolano, 2006].

Nuestro método de localización está basado en una búsqueda que evoluciona del grano grueso al grano fino. El mapa es dividido manualmente en seis submapas o categorías, correspondientes a seis entornos diferentes. El sistema almacena una base de datos de 721 imágenes, cubriendo completamente el entorno global, que identifican diferentes poses del robot en el mismo. En el paso de grano grueso se determina en qué submapa está localizado el robot. Este paso se apoya en una clasificación basada en los k vecinos más cercanos a partir de la información proporcionada por un conjunto de filtros simples aplicados a la imagen de entrada. La salida de este paso es el conjunto de k imágenes de la base de datos más similares a la imagen de entrada [Bonev et al., 2007a].

El paso de grano fino se basa en extracción de características; es en este paso donde aplicamos nuestro algoritmo de filtrado. El sistema extrae características salientes de la imagen; dichas características son descritas mediante descriptores SIFT. Un descriptor SIFT [Lowe, 2004] es un vector de 128 dimensiones que representa la apariencia de una característica saliente de la imagen, de manera que sea invariante a escala y rotación. El descriptor SIFT, y otros descriptores basados en SIFT, son usados comúnmente en aplicaciones de Visión Artificial debido a su mayor exactitud a la hora de producir emparejamientos correctos entre características [Mikolajczyk and Schmid, 2004a]. El siguiente paso

es emparejar las características extraídas de la imagen de entrada con aquellas extraídas de las k imágenes de la base de datos que fueron seleccionadas en el paso anterior. El algoritmo de emparejamiento se apoya en un criterio estructural que permite la eliminación de emparejamientos inconsistentes [Aguilar, 2006]. Finalmente, la salida del algoritmo de localización es aquella imagen de la base de datos, de entre las k escogidas por el paso de grano grueso, que produce el mayor número de emparejamientos con la imagen de entrada.

El objetivo de la búsqueda de grano grueso es doble. Por una parte permite seleccionar un conjunto reducido de candidatos; por lo tanto, no es necesario aplicar la búsqueda de grano fino a todas las imágenes de la base de datos. Y por otra parte, proporciona el identificador del submapa donde se encuentra localizado el robot. Con ello ya conocemos la categoría a la que pertenece la imagen de entrada, con lo que podemos utilizar un umbral previamente aprendido para filtrar píxeles de la imagen antes del emparejamiento estructural basado en extracción de características. En la Fig. 3.21 mostramos un resumen de los resultados de filtrado. Para cada uno de los seis entornos se indica el porcentaje medio de puntos filtrados por imagen. Usando el mínimo umbral válido, este porcentaje varía entre el 26% y el 60% para las diferentes categorías. También mostramos en la Fig. 3.23 algunos ejemplos de aplicación de nuestro algoritmo de filtrado a imágenes de este experimento de localización robótica. Cada fila muestra los resultados para una imagen diferente. En la parte izquierda podemos ver el resultado del algoritmo original de Kadir y Brady. En la parte derecha vemos el resultado del algoritmo Scale Saliency tras aplicar nuestro filtrado con el mínimo umbral T para la categoría correspondiente a cada imagen (las regiones en rojo muestran los puntos que fueron filtrados). Finalmente, en la parte derecha podemos ver el resultado de aplicar el algoritmo Scale Saliency tras aplicar nuestro filtrado usando $T = 0$ en todos los casos.

B.3.8 Conclusiones

En esta sección hemos presentado un algoritmo de filtrado que permite descartar píxeles de la imagen que probablemente no vayan a formar parte de sus características más salientes antes de aplicar el algoritmo Scale

Saliency. Nuestro algoritmo se basa en establecer un umbral de entropía normalizada en la escala máxima de estudio. Este umbral es inferido a partir de un conjunto de imágenes pertenecientes a la misma categoría o entorno. Realizamos experimentos para mostrar el efecto de diferentes parámetros del algoritmo en el resultado final y para demostrar su buen funcionamiento. También mostramos una aplicación práctica en el campo de la localización robótica. Las aportaciones de esta sección son:

- Estudiamos la evolución de la entropía de Shannon a lo largo del espacio de la imagen y del espacio de escalas. Demostramos que la entropía de un punto a escala máxima permite establecer un límite para la máxima saliencia de dicho punto a lo largo del espacio de escalas. Este hecho apoya la idea de que regiones homogéneas o no salientes en las escalas más altas también serán homogéneas o no salientes en las escalas más bajas.

- Propusimos un algoritmo simple de filtrado que no puede ser considerado factible: en el caso en el que se deseen evitar falsos positivos o negativos, el umbral tan solo puede ser establecido *después* de aplicar el propio algoritmo Scale Saliency. Sin embargo, esta primera aproximación sirve para demostrar que es posible filtrar una gran cantidad de puntos de la imagen utilizando únicamente información de la escala máxima. Además, este algoritmo constituye la base de nuestra solución final de filtrado.

- Demostramos que un conjunto de imágenes pertenecientes a una misma clase o entorno pueden ser agrupadas con el objetivo de aprender un umbral de filtrado que sea válido para nuevas imágenes pertenecientes a dicha clase o categoría.

- Indicamos como utilizar la Teoría de la Información para evaluar la aplicabilidad de nuestro algoritmo a una categoría de imágenes (por medio de la Información de Chernoff) y para establecer un rango de umbrales válidos para ella (mediante la divergencia de Kullback-Leibler). Nuestros experimentos demostraron que estas medidas, relacionadas entre sí, proporcionan una estimación de la cantidad de falsos negativos o positivos.

- La necesidad de conocer la clase a la que pertenece una imagen antes de aplicarle nuestro algoritmo puede parecer una gran limitación del mismo. Sin embargo, demostramos que puede ser utilizado en la práctica con un ejemplo de aplicación en el campo de la localización robótica. Otra posible aplicación podría ser la extracción de características y el seguimiento en vídeos de vigilancia. En este caso, el entorno al que pertenece la imagen es también conocido siempre.

Vemos dos posibles maneras de mejorar el método presentado en esta sección. La primera de ellas está relacionada con la división de imágenes en categorías. Durante nuestros experimentos mantuvimos las categorías de las bases de datos de imágenes *Object Category* y *Caltech101*, y también realizamos una división manual en submapas de las imágenes del experimento de localización robótica. En este último caso también intentamos agrupar las imágenes a partir de la información proporcionada por filtros simples, similares a los utilizados durante el paso de grano grueso del método de localización robótica. Las imágenes fueron representadas mediante vectores (construidos a partir de la salida de los filtros) y agrupados utilizando algoritmos conocidos de agrupamiento, como por ejemplo el método k-medias. A pesar de que el valor de la Información de Chernoff para estas categorías obtenidas de manera no supervisada fue mayor (de hecho, solo 5 categorías, y no 6, fueron necesarias para mejorar este valor), los resultados del filtrado en las imágenes de test fueron decepcionantes: la media de puntos filtrados y de tiempo ahorrado por imagen no fueron mejores que las obtenidas en el caso de dividir manualmente el mapa.

Por otra parte, nuestro trabajo podría ser el primer paso en el descubrimiento de nuevas propiedades de la entropía en el espacio de escalas, de tal forma que se pudieran diseñar nuevos filtros que permitieran descartar más puntos. Nuestra hipótesis es que dichos filtros podrían ser organizados en cascada, de tal forma que cada filtro procesara la salida del filtro del nivel superior, con lo que los filtros más complejos podrían ser dejados al final para que procesaran menos puntos (es algo similar a lo que se plantea en el método de reconocimiento de texto de Chen y Yuille [Chen and Yuille, 2004], que a su vez se basa en el método de reconocimiento de caras mediante Adaboost de Viola y

Jones [Viola and Jones, 2001]). Concretamente, el término de auto-disimilitud W_D podría ser analizado con tal fin. Este término de normalización pondera la entropía de un pixel dependiendo de la variación de la función de densidad de probabilidad local entre escalas (en nuestro caso, la variación del histograma de intensidades). Por lo tanto, los píxeles podrían ser descartados también en función de la distribución de intensidad de sus vecinos.

B.4 Scale Saliency multidimensional

B.4.1 Introducción

Los histogramas son utilizados durante el algoritmo Scale Saliency de Kadir y Brady [Kadir and Brady, 2001] como herramienta para la estimación de entropía y de auto-similitud entre escalas (ver Ec. B.2 y Ec. B.4). Esta estimación basada en histogramas está sujeta a la maldición de la dimensionalidad: la complejidad espacial y temporal crece exponencialmente conforme lo hace la dimensionalidad de los datos. Esto quiere decir que el algoritmo de Kadir y Brady es lento cuando es aplicado a imágenes en color, y que no es factible utilizarlo en el caso de dimensionalidades aun mayores. Además, conforme las dimensiones del histograma aumentan, los datos son más y más dispersos; la consecuencia de esto es que los histogramas son menos informativos.

El objetivo de esta Sección es la creación de una versión del algoritmo Scale Saliency que pueda trabajar con datos multidimensionales sin estas restricciones. Analizamos estimadores alternativos de medidas de la Teoría de la Información cuya complejidad conforme aumenta el número de dimensiones es lineal, y probamos su aplicación al algoritmo Scale Saliency, en términos de eficiencia computacional, y también en términos de la calidad y la cantidad de las características extraídas.

En primer lugar, en la Sección B.4.2, analizamos diferentes algoritmos de estimación basados en Árboles de Longitud Mínima y Grafos de K-Vecinos más Cercanos. A continuación, en la Sección B.4.3, resumimos un método alternativo de estimación conocido como k-d partition, que divide recursivamente los datos siguiendo el mismo esquema que el algoritmo k-d tree. Proponemos también una nueva medida de divergencia basada

en k-d partition y en la distancia de variación total. En la Sección B.4.4 presentamos nuestra versión multidimensional del algoritmo Scale Saliency. Este algoritmo se construye a partir de los estimadores basados en el método k-d partition, debido a las conclusiones extraídas de nuestros experimentos en la Sección B.4.5. En estos experimentos comprobamos la complejidad computacional y el error de estimación de los diferentes métodos de estimación presentados en esta Sección, y también comprobamos la cantidad y la calidad de las características extraídas cuando se aplican al algoritmo Scale Saliency. Finalmente, en la Sección B.4.10, mostramos una aplicación de nuestra versión del algoritmo Scale Saliency al problema de la categorización de texturas.

B.4.2 Estimación de entropía y divergencia a partir de grafos entrópicos

El trabajo de Beirlant *et al.* [Beirlant et al., 2001] proporciona una buena visión general sobre los métodos no paramétricos de estimación de entropía. Los autores dividen estos métodos en dos grupos: métodos basados en sustitución (plug-in) y métodos basados en la distancia entre muestras. Los primeros consisten en estimar la distribución de los datos e introducir dicha estimación en la fórmula del cálculo de la entropía. La estimación de entropía basada en histogramas es un buen ejemplo de método de estimación basado en sustitución. Otros métodos se basan en el uso de la distribución basada en núcleos, como las ventanas de Parzen [Erdogmus et al., 2004]. El segundo grupo de estimadores se basan en el cálculo de la distancia entre muestras. Este es el caso por ejemplo del estimador de Leonenko *et al.* [Leonenko et al., 2008] y la estimación de la α-entropía de Rényi a partir de Árboles de Longitud Mínima de Hero *et al.* [Hero and Michel, 1999] (estos dos últimos métodos son explicados más adelante).

De todos estos métodos escogemos aquellos basados en grafos. Su convergencia asintótica es más rápida, especialmente en el caso de datos de alta dimensionalidad que no siguen una distribución suave [Hero et al., 2003]. Los métodos basados en núcleos (como la estimación basada en ventanas de Parzen) proporcionan peores resultados en estos casos y son más sensibles a la presencia de valores atípicos.

Además, al igual que en el caso de los histogramas, los métodos basados en núcleos se ven afectados por la maldición de la dimensionalidad: mayores dimensionalidades requieren un aumento considerable del número de muestras para poder realizar una buena estimación. En esta sección utilizaremos el término *grafos entrópicos* para referirnos a la familia de grafos mínimos que abarcan un conjunto de muestras y que producen una estimación consistente de la entropía [Hero et al., 2002]. Concretamente, usamos estimadores basados en Árboles de Longitud Mínima (MSTs o Minimal Spanning Trees) y Grafos de K-Vecinos más Cercanos (KNNGs o K-Nearest Neighbour Graphs) [Costa and Hero, 2004].

Nuestra versión multidimensional del algoritmo Scale Saliency se aplica a aquellas imágenes en las que cualquier píxel $\mathbf{x}_i \in X$ se representa mediante un vector de d dimensiones (vectores de 3 dimensiones en el caso de imágenes RGB, por ejemplo, o vectores de 31 dimensiones en el caso de imágenes hiperespectrales con 31 bandas). En nuestros experimentos de estimación de entropía a partir de grafos entrópicos, la vecindad $R_{\mathbf{x}}$ de un píxel se representa como un grafo no dirigido y totalmente conexo $G = (V, E)$, siendo los nodos $\mathbf{v}_i \in V$ vectores de d dimensiones que se corresponden con los píxeles $\mathbf{x}_i \in R_{\mathbf{x}}$, y siendo E el conjunto de aristas que conectan cada par de nodos. El peso de cada arista en E será la distancia euclídea en $\mathcal{R}^d$ entre los dos nodos incidentes en dicha arista. El MST de G es el árbol [4] de distancia mínima total entre todos los árboles que se pueden construir a partir de G. Si $|V| = n$, entonces $|E| = n - 1$. Por otra parte, el KNNG construido a partir de G será el subgrafo de G que conecta cada nodo con sus k vecinos más cercanos. En este caso, $|E| = kn$. En la Fig. 4.1 se puede observar un ejemplo de cada uno de estos dos tipos de grafo.

Algoritmos de construcción de grafos entrópicos

Es posible encontrar muchos métodos en la literatura para la construcción de MSTs y KNNGs. De hecho, los primeros algoritmos de construcción de MSTs datan de la década de los veinte (aunque un poco anticuado, el estudio del estado del arte de Eisner [Eisner, 1997] es un buen punto de inicio para

[4] un árbol será un subgrafo de G que conectará cada par de nodos (v_i, v_j) a través de exactamente una única ruta

empezar a estudiar este tipo de algoritmos). En esta sección resumimos los algoritmos de construcción de MSTs y KNNGs utilizados durante los experimentos de la Sección B.4.5.

En primer lugar describimos dos algoritmos clásicos para la construcción de MSTs: el algoritmo de Kruskal [Kruskal, 1956] y el de Prim [Prim, 1957]. En el primero las aristas son ordenadas según su peso (distancia Euclídea en nuestro caso). A continuación las aristas se van tomando en orden para comprobar si forman parte o no del MST. Si el resultado de añadir la arista es la creación de un ciclo, dicha arista se descarta. En otro caso, se mantiene. El segundo algoritmo también comienza sin aristas en el MST. pero en lugar de mantener un conjunto de árboles que terminan uniéndose en un único MST, consiste en el crecimiento de un solo árbol. El algoritmo comienza seleccionando la arista de menor peso que incida sobre un nodo inicial arbitrario. A continuación, en cada paso, se selecciona la arista de menor peso que parta del árbol obtenido hasta ese momento. En ambos casos usamos una estructura de datos de tipo *pairing heap* [Fredman et al., 1986] para seleccionar las aristas en orden de peso. La implementación de esta estructura de datos es simple, y aunque no se ha podido estimar su complejidad temporal, en la práctica este algoritmo se ha mostrado más rápido que otros algoritmos eficientes de este tipo [Stasko, 1987][Moret and Shapiro, 1991].

El algoritmo de Borůvka [Borůvka, 1926] es un buen ejemplo de algoritmo de enfoque ascendente que es fácilmente paralelizable. El primer paso es añadir al MST la arista de menor peso que parta de cada nodo. Cada componente conexa de nodos formará un subárbol. A continuación, en cada iteración posterior, la arista de menor peso que parta de cada subárbol se añadirá al MST, hasta que se construya el árbol definitivo. El algoritmo incluye un paso de compresión, en el que aquellas aristas que no partan de un nodo terminal del árbol son eliminadas. En la práctica estos tres algoritmos (Borůvka, Prim y Kruskal) son los más utilizados. Sin embargo, al menos teóricamente, existen otros algoritmos de menor complejidad, próxima incluso a la complejidad lineal (ver [Fredman and Tarjan, 1987], [Gabow et al., 1986] y [Pettie and Ramachandran, 2002]). Sin embargo, su implementación es compleja y son considerados algoritmos poco prácticos dado al efecto de los factores constantes en el tiempo de ejecución [Katriel et al., 2003].

El coste temporal del algoritmo de Katriel *et al.* [Katriel et al., 2003] es similar al del algoritmo de Fredman y Tarjan [Fredman and Tarjan, 1987], pero en la práctica su implementación es más sencilla y su tiempo de ejecución es menor. El primer paso consiste en construir un subgrafo G' a partir de G tomando un conjunto aleatorio de aristas. A continuación, el algoritmo calcula el Bosque de Longitud Mínima (MSF o Minimal Spanning Forest) a partir de G' (ya que G' puede que no sea totalmente conexo), al que llamaremos T'. El MSF es el conjunto de árboles de menor peso que se extiende por todos los nodos de cada componente conexa de un grafo. En el siguiente paso, el algoritmo elimina las aristas $e \in E$ de G que sean la arista de mayor peso en cualquier ciclo del grafo formado por $T' \bigcup \{e\}$. Finalmente, el algoritmo obtiene el MST a partir de G utilizando las aristas que no hayan sido eliminadas. En nuestra implementación, tanto el MSF como el MST se construyen por medio del algoritmo de Prim basado en la estructura de datos *pairing heap*. Este es el único algoritmo con componente aleatoria que utilizamos en nuestros experimentos.

Con respecto a la construcción de un KNNG, básicamente nos limitaremos a encontrar los k vecinos más cercanos de cada nodo del grafo. Para ello hacemos uso de la versión semidinámica del algoritmo k-d tree de Bentley y Friedman [Bentley, 1990]. El algoritmo k-d tree es ampliamente utilizado y también es eficiente [Bentley, 1975][Friedman et al., 1975], pero su mayor inconveniente es que no se trata de un algoritmo adecuado para altas dimensionalidades. El número de nodos n debería ser $n >> d$; en caso contrario la mayoría de los nodos del árbol son evaluados durante la operación de búsqueda del vecino más cercano y la eficiencia del proceso sería equivalente a la de una búsqueda exhaustiva [Goodman and O'Rourke, 2004]. Podría parecer contradictorio su uso en nuestros experimentos, ya que es precisamente la maldición de la dimensionalidad lo que tratamos de evitar; sin embargo el uso de un k-d tree semidinámico nos va a permitir la exploración del espacio de escalas mientras se va actualizando el árbol. El efecto de esto sobre el tiempo de ejecución es notable, comparable al efecto de utilizar histogramas acumulativos en la implementación del algoritmo Scale Saliency de Kadir y Brady, en el que el histograma para una determinada escala se utiliza como base para construir el histograma para la siguiente. Durante nuestra

investigación no encontramos ningún algoritmo dinámico de construcción de MSTs que pudiéramos aplicar a nuestro problema. Los algoritmos más eficientes de este tipo que encontramos o bien requerían el uso de grafos planares [Eppstein et al., 1996] o bien nodos con un grado máximo de dos [Frederickson, 1983].

El algoritmo semidinámico de construcción de k-d trees de Bentley [Bentley, 1990] permite el borrado y la restauración de nodos, pero no la inserción de nodos nuevos. Estas operaciones, junto con la de la búsqueda del vecino más cercano, tienen coste constante. Además, la estructura de datos utilizada se ve menos expuesta a la aparición de casos degenerados. La versión semidinámica del algoritmo k-d tree añade un nuevo campo llamado *vacío* a la estructura de datos que representa un nodo del árbol. Si durante la búsqueda del vecino más cercano de un nodo se alcanza un nodo vacío, éste es inmediatamente abandonado. Para borrar una muestra almacenada en el árbol se elimina de su correspondiente nodo hoja en el k-d tree y a continuación se viaja desde las hojas a la raíz del árbol activando el campo *vacío* cuando sea necesario. Los experimentos de Bentley demuestran que no es necesario volver a balancear el árbol al realizar esta operación, ya que los nodos borrados no afectan en gran medida al tiempo de ejecución de las operaciones de búsqueda. La operación de restauración es similar. La muestra se vuelve a asignar a su correspondiente nodo hoja, y el algoritmo visita los nodos desde dicho nodo hoja a la raíz desactivando el campo *vacío* cuando sea necesario.

Al incluir la versión semidinámica de la estructura k-d tree en el algoritmo Scale Saliency multidimensional el espacio de escalas debe ser explorado desde s_{max} hasta s_{min}. Tras estimar la entropía del punto $\mathbf{x}$ en la escala s, aquellos nodos que no sean parte de $R_{\mathbf{x}}$ en la escala $s-1$ son borrados antes de estimar la entropía para dicha escala más baja. La complejidad temporal de la exploración del espacio de escalas desde s_{min} a s_{max} sería menor, pero no conocemos ningún algoritmo de este tipo que permita la inserción de nuevos puntos.

Estimación de la α-entropía de Rényi

La α-entropía de Rényi [Rényi, 1961] es una generalización de la entropía de Shannon, y por lo tanto se trata de una medida de la incertidumbre asociada a una variable aleatoria. La α-entropía de una variable aleatoria X se define como:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^\alpha \right) \quad , \tag{B.16}$$

donde p_i son las probabilidades de los distintos valores aleatorios x_i de X. Hero y Michel propusieron un método de estimación de la α-entropía de Renyi a partir de MSTs [Hero and Michel, 1999] que también puede ser aplicado en el caso de KNNGs [Costa and Hero, 2004]. Se trata de un método sencillo basado en la longitud del grafo que se expande a través de todos los nodos: mayor distancia entre las muestras conlleva mayor incertidumbre, y por lo tanto, mayor entropía (Fig. 4.2). En un espacio de d dimensiones, siendo $d \geq 2$, la estimación de la α-entropía viene dada por

$$H_\alpha(X_n) = \frac{d}{\gamma} \left[\ln \frac{L_\gamma(X_n)}{n^\alpha} - \ln \beta_{L_{\gamma,d}} \right] \quad . \tag{B.17}$$

En esta ecuación, γ depende de α y de la dimensionalidad de los datos ($\alpha = (d-\gamma)/d$), y el factor de corrección $\beta_{L_{\gamma,d}}$ depende del criterio de minimización del grafo (esto es, del tipo de grafo de longitud mínima construido – MST, KNNG, etc.), pero no depende de la distribución de probabilidad de las muestras. Este factor podría ser aproximado por medio de (i) una simulación de Monte Carlo a partir de muestras distribuidas aleatoriamente en el cubo unitario $[0,1]^d$ y (ii) una estimación para valores altos de d dada por $(\gamma/2)ln(d/(2\pi e))$ [Bertsimas and van Ryzin, 1990]. En nuestros experimentos ignoramos este factor. Con respecto a $L_\gamma(X_n)$, se trata de la longitud ponderada de las aristas $\{e\}$ del MST o del KNNG

$$L_\gamma(X_n) = \min_{M(X_n)} \sum_{e \in M(X_n)} |e|^\gamma \quad , \tag{B.18}$$

donde $M(X_n)$ representa cualquier combinación de aristas que se expanda por todos los nodos del grafo (con lo que el mínimo $M(X_n)$ denota un MST o un KNNG, dependiendo del método que se esté empleando para

la construcción del grafo mínimo), $X_n = \{x_1, \ldots, x_n\}$ es el conjunto de vértices, $\{e\}$ el conjunto de aristas, $|e|$ la longitud de una determinada arista, y $0 < \gamma < d$.

La estimación de la α-entropía a partir de grafos es tan solo posible para $0 \leq \alpha \leq 1$. La α-entropía converge a la entropía de Shannon cuando $\alpha \to 1$, pero no es posible establecer el valor $\alpha = 1$ (como se puede apreciar en la Ec. B.17). En [Peñalver et al., 2006] se aproxima el valor de H_α para $\alpha = 1$ (H_1) por medio de una función continua que captura la tendencia de H_α en la vecindad del valor 1. Este valor a^* que aproxima H_1 viene dado por

$$\alpha^* = 1 - \frac{1.271 + 1.3912e^{-0.2488d}}{n} \quad . \tag{B.19}$$

A pesar de disponer de esta estimación de la entropía de Shannon a partir de la α-entropía de Renyi, no hicimos uso de ella durante nuestros experimentos, debido a las siguientes razones:

- La aproximación de la Ec. B.19 se obtuvo experimentalmente a partir de distribuciones Gaussianas con $2 \leq d \leq 5$. Sin embargo, la información multidimensional obtenida a partir de una imagen no suele seguir una distribución Gaussiana. Además, en nuestros experimentos llegamos a aplicar el algoritmo Scale Saliency a datos de 31 dimensiones (ver Sección B.4.5).

- Nuestros experimentos demostraron que la Ec. B.17 es inestable en el caso de imágenes multidimensionales, si se utilizan MSTs o KNNGs con un valor de k bajo. La principal causa de esta inestabilidad es la presencia de aristas de longitud cero, cuya consecuencia podría ser la aparición de grafos entrópicos de baja longitud, siendo $L_\gamma(X_n) < n^\alpha$, y por lo tanto, obteniéndose valores negativos de entropía. Una posible solución, inspirada en la aportada por Neemuchwala *et al.* [Neemuchwala et al., 2006] para solventar la inestabilidad de la estimación de la α-Información Mutua, es añadir ruido uniforme a los valores de los píxeles. Sin embargo, este método afecta a los resultados de la estimación.

El método de Leonenko de estimación de entropía

En esta sección tratamos la estimación de entropía a partir de la distancia de los k-vecinos más cercanos propuesta por Leonenko *et al.* [Leonenko et al., 2008]. El objetivo de su trabajo era definir un estimador para la entropía de Rényi [Rényi, 1961]

$$H_\alpha^* = \frac{1}{1-\alpha} \log \int f^\alpha(x) dx, \alpha \neq 1 \ , \tag{B.20}$$

o de la entropía de Havrda y Charvát [Havrda and Charvát, 1967], también conocida como entropía de Tsallis [Tsallis, 1988]

$$H_\alpha = \frac{1}{\alpha-1} \left(1 - \int f^\alpha(x) dx\right), \alpha \neq 1 \ , \tag{B.21}$$

a partir de un conjunto de N muestras independientes e idénticamente distribuidas. Para ello se basaron en un trabajo anterior de Kozachenko y Leonenko [Kozachenko and Leonenko, 1987]. Estos autores afirman que H_α^* es una función de H_α estrictamente creciente, por lo que la maximización de H_α también lleva a la maximización de H_α^*. Cuando α tiende a 1, tanto H_α como H_α^* tienden a la entropía de Shannon:

$$H_1 = -\int f(x) \log f(x) dx \ . \tag{B.22}$$

La estimación de la Ec. B.22 a partir de N muestras independientes e idénticamente distribuidas es:

$$\hat{H}_{N,k} = \frac{1}{N} \sum_{i=1}^{N} \log \xi_{N,i,k} \ , \tag{B.23}$$

siendo

$$\xi_{N,i,k} = (N-1) e^{-\psi(k)} V_d (\rho_{k,N-1}^{(i)})^d \ . \tag{B.24}$$

En la anterior ecuación V_d es el volumen de la esfera unitaria de d dimensiones, $\rho_{k,N-1}^{(i)}$ es la distancia al k-ésimo vecino más cercano de i cuando se consideran el resto de las $N-1$ muestras, y

$$\psi(z) = \frac{\Gamma'(z)}{\Gamma(z)} \tag{B.25}$$

es la función *digamma*. Esta función es $\psi(1) = -\gamma$, siendo $\gamma \simeq 0,5772$ la constante de Euler, y $\psi(k) = -\gamma + A_{k-1}$ para cualquier entero $k \geq 1$. Finalmente, $A_0 = 0$ y

$$A_j = \sum_{i=1}^{j} 1/i \ . \tag{B.26}$$

El estimador mostrado en la Ec. B.23 requiere la búsqueda de los k vecinos más cercanos de cada muestra. En el caso de nuestro algoritmo Scale Saliency basado en grafos, construimos el KNNG de las muestras en $R_{\mathbf{x}}$ siguiendo el método semidinámico explicado en la Sec. B.4.2. Este algoritmo mantiene una estructura de datos que permite acceder eficientemente a los vecinos más cercanos y actualizar el KNNG en el rango de escalas, desde s_{max} a s_{min}. Se puede ver un ejemplo de aplicación en la Fig. 4.3.

Estimación de la divergencia de Henze-Penrose mediante el test de Friedman-Rafsky

Hasta ahora hemos visto algunos ejemplos de estimadores de entropía que resuelven el principal problema de los histogramas en el caso de datos multidimensionales: su complejidad espacial y temporal exponencial con respecto al número de dimensiones. Estos estimadores están basados en grafos y no requieren la estimación previa de la función de densidad de probabilidad de las muestras. Sin embargo, en el algoritmo Scale Saliency nos encontramos con otro paso que también depende de la estimación de dicha densidad de probabilidad: los picos de entropía en el espacio de escalas deben ser ponderados por medio de una medida de auto-similitud, cuyo valor debe encontrarse en el rango $[0, 1]$ (Ec. B.4). Esta ponderación penaliza a aquellas características que son salientes en un rango de escalas amplio (se debe recordar que el objetivo del algoritmo es detectar características que sean salientes en un rango estrecho de escalas). En nuestra propuesta de algoritmo Scale Saliency basada en grafos sustituimos esta medida por una estimación de la divergencia de Henze-Penrose. La divergencia de Henze-Penrose [Henze and Penrose, 1999] entre dos distribuciones f y g viene dada por:

$$D_{HP}(f||g) = \int \frac{p^2 f^2(z) + q^2 g^2(z)}{pf(z) + qg(z)} dz \ , \tag{B.27}$$

donde $p \in [0,1]$ y $q = 1 - p$. Esta divergencia es el límite del estadístico de Friedman-Rafsky [Friedman and Rafsky, 1979], que a su vez es una generalización multidimensional basada en MSTs del test de Wald-Wolfowitz. El test de Wald-Wolfowitz mide la divergencia entre dos distribuciones f_X y f_O en $\mathcal{R}^d$ cuando $d = 1$, a partir de dos conjuntos de n_x y n_o muestras, respectivamente. En primer lugar, las $n = n_x + n_o$ muestras se ordenan en orden ascendente, y se etiquetan como X ó O dependiendo de la distribución a la que pertenezcan. El test se basa en el número R de secuencias de muestras consecutivas con la misma etiqueta:

$$W = \frac{R - \frac{2n_o n_x}{n} - 1}{\left(\frac{2n_x n_o (2n_x n_o - n)}{n^2 (n-1)}\right)^{\frac{1}{2}}} \ . \tag{B.28}$$

Las dos distribuciones se consideran similares si el valor de R es bajo (lo cual producirá a su vez un valor bajo de W). Este test es consistente cuando n_x/n_o no es cercano ni a 0 ni a ∞, y cuando $n_x, n_o \to \infty$. El test de Friedman-Rafsky generaliza el test de Wald para $d > 1$, debido al hecho de que los MSTs y los KNNGs relacionan muestras cercanas en $\mathcal{R}^d$. Sean $X = \{x_i\}$ y $O = \{o_i\}$ dos conjuntos de muestras obtenidas respectivamente a partir de f_X y f_O. Los pasos del test de Friedman-Rafsky son:

1. construir el MST para el conjunto de muestras $X \bigcup O$.
2. eliminar las aristas que no conecten una muestra de X con una muestra de O.
3. la proporción de aristas no eliminadas converge a 1 menos la divergencia de Henze-Penrose (Ec. B.27) entre f_X y f_O.

La Fig. 4.4 muestra un ejemplo de uso. La aplicación del test de Friedman-Rafsky al algoritmo Scale Saliency basado en grafos entrópicos es sencilla. Sea s la escala en la que se encontró un pico de entropía. Para ponderar este valor de entropía se debe medir la disimilitud con respecto a la escala $s - 1$ (Ec. B.4). Sean $M(X_{m,s})$ y $M(X_{n,s-1})$ los grafos entrópicos

construidos para estimar la entropía en las escalas s y $s-1$ (con lo que $m > n$). La ventaja del test de Friedman-Rafsky es que no es necesario construir un nuevo MST o KNNG, ya que $X_{n,s-1} \subset X_{m,s}$ (conforme aumentamos de escala, nuevos píxeles se añaden a $R_{\mathbf{x}}$, pero nunca se eliminan). Por lo tanto, $M(X_{m,s})$ es el grafo entrópico de longitud mínima para $X_{n,s-1} \bigcup X_{m,s}$. Durante el paso de ponderación tan solo se deberá contar el número de aristas de $M(X_{m,s})$ que conectan un nodo de $X_{m,s}/X_{n,s-1}$ con un nodo de $X_{n,s-1}$.

B.4.3 Estimación de entropía y divergencia mediante el algoritmo k-d partition

Los resultados de nuestros experimentos (Sec. B.4.5) demostraron que la mayor desventaja a la hora de usar los métodos de estimación basado en grafos entrópicos es el alto coste temporal asociado a la construcción de las estructuras de datos necesarias. Este gran coste computacional es debido en gran medida a la necesidad del cálculo de distancias entre muestras. Stowell y Plumbley propusieron un nuevo estimador de la entropía que resolvía este problema [Stowell and Plumbley, 2009]. En su algoritmo la entropia es también estimada a partir del espaciado entre muestras, pero sin la necesidad de calcular ninguna distancia. El método está basado en el paso de partición del espacio llevado a cabo durante el algoritmo k-d tree. Por lo tanto, lo que hace su estimador es dividir el espacio de las muestras en un conjunto de celdas a partir de las cuales se estima su entropía.

Sea X una variable aleatoria que toma valores de d dimensiones, y sea $f(x)$ su función de densidad de probabilidad asociada. Sea $A = \{A_j | j = 1, \ldots, m\}$ una partición de X para la que $A_i \cap A_j = \emptyset$ si $i \neq j$ y que cumple $\bigcup_j A_j = X$. Entonces, la aproximación en cada celda de $f(x)$ viene dada por

$$f_{A_j} = \frac{\int_{A_j} f(x)}{\mu(A_j)} \ , \tag{B.29}$$

donde $\mu_(A_j)$ es el volumen en d dimensiones de A_j. Si $f(x)$ es desconocida, pero disponemos de un conjunto de muestras de la misma $X = \{x_1, \ldots, x_n\}$, siendo $x_i \in \mathcal{R}^d$, es posible aproximar la probabilidad de $f(x)$ en cada celda como $p_j = n_j/n$, donde n_j es el número de muestras en la celda A_j. Por lo tanto

$$\hat{f}_{A_j}(x) = \frac{n_j}{n\mu(A_j)} \ , \tag{B.30}$$

siendo $\hat{f}_{A_j}(x)$ un estimador consistente de $f(x)$ cuando $n \rightarrow \infty$ [Breiman et al., 1984][Zhao et al., 1990]. Si la entropía diferencial viene dada por

$$H = -\int f(x) \log f(x) dx \ , \tag{B.31}$$

entonces es posible sustituir en la anterior ecuación $f(x)$ (según Ec. B.30) para obtener la estimación del conjunto total de muestras dada una partición A:

$$\hat{H} = \sum_{j=1}^{m} \frac{n_j}{n} \log \left(\frac{n}{n_j} \mu(A_j) \right) \ . \tag{B.32}$$

La partición del espacio de muestras se realiza recursivamente siguiendo el método de partición del algoritmo de construcción de k-d trees. En cada nivel las muestras son divididas según su mediana en un determinado eje. Después el particionado es aplicado a cada subespacio recursivamente hasta que se cumple cierto criterio de uniformidad. El objetivo de este criterio es que se produzcan celdas con una distribución empírica uniforme para obtener la mejor estimación posible de $f(x)$. La prueba de uniformidad escogida es eficiente, depende de la mediana, y viene dada por

$$Z_j = \sqrt{n_j} \frac{2med_d(A_j) - min_d(A_j) - max_d(A_j)}{max_d(A_j) - min_d(A_j)} \ , \tag{B.33}$$

donde $med_d(A_j)$, $min_d(A_j)$ y $max_d(A_j)$ son respectivamente la mediana, el mínimo y el máximo de los valores en la celda A_j en la dimensión D. Stowell y Plumbley consideran que un valor $|Z_j| > 1.96$ indica una gran falta de uniformidad de los datos, y por lo tanto que dicha celda debería ser dividida más. Por otra parte, para conseguir un nivel de ramificación adecuado, se incluye una heurística adicional. En concreto, el algoritmo no aplica la prueba de uniformidad anterior hasta que no se haya alcanzado el nivel dado por

$$L_n = \left\lceil \frac{1}{2} \log_2(n) \right\rceil \ . \tag{B.34}$$

A la hora de implementar el algoritmo debemos considerar dos factores. En primer lugar, en el algoritmo de Stowell y Plumbley el eje (dimensión)

usado para dividir las muestras es escogido secuencialmente en cada llamada recursiva. Los autores afirman que para obtener una estimación consistente, la cantidad de muestras requerida aumenta exponencialmente con la dimensión. Durante el algoritmo Scale Saliency no se satisface este requisito. Con el objetivo de minimizar el efecto de un bajo número de muestras, la dimensión escogida en cada llamada recursiva podría ser aquella para la que existiera una máxima varianza. Esta heurística se aplica también normalmente durante el algoritmo de construcción de un k-d tree [5].

En segundo lugar comentamos algo a tener en cuenta con respecto al soporte de los datos. El algoritmo de particionado que acabamos de resumir puede producir celdas con volumen infinito y/o sin ninguna muestra que no pueden ser aplicadas en la Ec. B.32. Stowell y Plumbley proponen acotar los límites de las celdas a un volumen finito definido por el soporte de cada celda, estimado a partir de los máximos y mínimos de sus muestras en cada dimensión. Esta solución, por supuesto, afecta a los resultados de la estimación de la Ec. B.32, pero en menor medida que ignorar completamente las celdas de volumen infinito.

Estimación de la divergencia a partir de k-d partition

Durante el algoritmo Scale Saliency basado en grafos entrópicos podemos usar, como se indicó anteriormente en la Sec. B.4.2, el test de Friedman-Rafsky para la ponderación de los picos de entropía: permite reutilizar los grafos construidos durante la estimación de la entropía, es simple y proporciona valores en el rango $[0, 1]$ (como la medida de disimilitud en el algoritmo Scale Saliency de Kadir y Brady [Kadir and Brady, 2001]). En la sección anterior presentamos otro método para la estimación de entropía a partir de una distribución multidimensional, cuya complejidad no aumenta exponencialmente con la dimensionalidad de los datos. Si deseamos aplicar este nuevo método como parte del algoritmo Scale Saliency, necesitaremos definir un nuevo estimador de la divergencia entre dos distribuciones que se base en k-d partition. El autor de esta tesis no conoce ningún algoritmo de

[5] Sin embargo, como se puede ver en los resultados de nuestros experimentos en la Sec. B.4.5, aplicar esta heurística en el algoritmo Scale Saliency basado en k-d partition empeora la calidad de las características extraídas.

estimación de divergencia de este tipo, por lo que en esta sección se definirá uno nuevo.

Nuestra nueva divergencia basada en el algoritmo k-d partition está inspirada en la distancia de variación total [Denuit and Bellegem, 2001], aunque podría ser interpretada también como una distancia L1. La distancia de variación total entre dos distribuciones de probabilidad P y Q en una σ-álgebra F[6] viene dada por

$$sup\{|P(X) - Q(X)| : X \in F\} \ . \tag{B.35}$$

En el caso de un alfabeto finito, la distancia de variación total es

$$\delta(P, Q) = \frac{1}{2} \sum_{x} |P(x) - Q(x)| \ . \tag{B.36}$$

Sean $f(x)$ y $g(x)$ dos distribuciones, a partir de las cuales se extrajeron respectivamente un conjunto X de n_x muestras y un conjunto O de n_o muestras. Si aplicamos el esquema de particionado del algoritmo k-d partition a todas las muestras de $X \bigcup O$ el resultado será una partición A de $X \bigcup O$, siendo $A = \{A_j | j = 1 \ldots, p\}$ (ver Sección B.4.3). En el caso de $f(x)$, la probabilidad de cualquier celda A_j se podría definir como

$$p(A_j) = \frac{n_{x,j}}{n_x} = p_j \ , \tag{B.37}$$

donde $n_{x,j}$ es el número de muestras de X en la celda A_j. De la misma forma, en el caso de $g(x)$ la probabilidad de cada celda A_j se puede definir como

$$p(A_j) = \frac{n_{o,j}}{n_o} = q_j \ , \tag{B.38}$$

donde $n_{o,j}$ es el número de muestras de O en la celda A_j. Dado que ambos conjuntos de muestras comparten la misma partición A, se podría considerar que el conjunto de celdas A_j forma un alfabeto finito, con lo cual podríamos calcular la distancia de variación total entre $f(x)$ y $g(x)$ como

$$D(O||X) = \frac{1}{2} \sum_{j=1}^{p} |p_j - q_j| \ . \tag{B.39}$$

[6]Una σ-álgebra sobre un conjunto X es una colección no vacía de subconjuntos de X (incluyendo a X) que está cerrada bajo uniones contables y complementos.

Esta medida de distancia satisface $0 \leq D(O||X) \leq 1$, por lo que podría ser utilizada como medida de auto-similitud durante el algoritmo Scale Saliency. El valor mínimo $D(O||X) = 0$ se obtiene cuando todas las celdas A_j contienen la misma proporción de muestras de X y de O. Por otra parte, el valor máximo $D(O||X) = 1$ se obtiene cuando todas las muestras en cualquier celda A_j provienen de una única distribución. En la Fig. 4.5 mostramos dos ejemplos de estimación de divergencia utilizando esta medida, para el caso de dos conjuntos de muestras obtenidas a partir de dos distribuciones Gaussianas. En el caso de la izquierda ambas Gaussianas tienen la misma media y desviación típica (en este caso $D(O||X) = 0.24$) mientras que en el caso de la derecha las dos Gaussianas tienen medias diferentes, por lo que en la mayoría de las celdas encontraremos muestras pertenecientes a una única distribución (en este caso $D(O||X) = 0.92$).

B.4.4 Scale Saliency multidimensional: el algoritmo

Nuestra versión multidimensional del algoritmo Scale Saliency se muestra en el Algoritmo 2. Esta versión hace uso del método de estimación de entropía conocido como k-d partition y también de nuestra nueva medida de divergencia (Sección B.4.3) para generalizar el algoritmo original [Kadir and Brady, 2001]. La entrada del algoritmo es un conjunto de m características extraídas para cada píxel de la imagen. Algunos ejemplos de estas características podrían ser la orientación y magnitud del gradiente, su color (como por ejemplo la intensidad de las tres bandas RGB), la intensidad del píxel a lo largo de un conjunto de fotogramas en una secuencia de vídeo o la salida de un banco de filtros aplicado al píxel. El algoritmo es simple, y tan solo es necesario hacer un comentario acerca del cálculo de la divergencia entre escalas (línea 4). En este paso, debido a que $X_{i-1} \subset X_i$, sería posible utilizar la partición realizada durante la estimación de entropía para calcular $D(X_{i-1}||X_i)$. La salida del algoritmo es un conjunto de valores de saliencia almacenados en una matriz HW que debe ser reordenada: las características más salientes de la imagen serán aquellas con un mayor valor de saliencia.

Este mismo algoritmo podría aplicarse con medidas de estimación de entropía y divergencia basadas en grafos entrópicos, ya sean KNNGs o MSTs (ver Secciones B.4.2 y B.4.2). Sin embargo, nosotros proponemos la versión

Input: Una matriz de m dimensiones I conteniendo m características para cada píxel de la imagen

Output: Una matriz HW conteniendo un valor de entropía ponderada para todos los píxeles en todas las escalas que se correspondan a picos de entropía

foreach *píxel x de la imagen* **do**
  foreach *escala s_i entre s_{min} y s_{max}* **do**
    (1) Crear un conjunto de muestras de m dimensiones $X_i = \{x_i\}$ a partir de la vecindad local del píxel x en I en la escala s_i; (2) Aplicar el algoritmo k-d partition a X para estimar la entropía $H(s_i)$
    if $i > s_{min} + 1$ **then**
      if $H(s_{i-2}) < H(s_{i-1}) > H(s_i)$ **then**
        `(* Pico de entropía *)`
        (4) divergencia basada en k-d partition: $W = D(X_{i-1}||X_{i-2})$;
        (5) $HW(s_{i-1}, x) = H(s_{i-1}) \cdot W$;
      end
      else
        (6) $HW(s_{i-1}, x) = 0$;
      end
    end
  end
end

Algorithm 2: El algoritmo Scale Saliency multidimensional

basada en k-d partition debido a que los resultados experimentales de la Sección B.4.6 muestran que este algoritmo es mucho más rápido. Además, en la Sección B.4.8 también demostramos que la calidad de las características extraídas en ambos casos es similar, por lo que el uso de grafos entrópicos no tiene ninguna ventaja sobre el uso de k-d partition.

B.4.5 Resultados experimentales

En esta sección analizamos experimentalmente los dos tipos de métodos de estimación: grafos y k-d partition. Comparamos ambas soluciones en términos de eficiencia computacional, error en la estimación de la entropía y la divergencia, y calidad y número de las características salientes extraídas. El objetivo de estos experimentos es seleccionar el mejor de estos métodos para poder ser aplicado al algoritmo Scale Saliency (ver arriba).

B.4.6 Tiempo de ejecución

El objetivo de esta sección es doble. Por una parte comparamos los tiempos de ejecución de los algoritmos de construcción de MSTs y KNNGs descritos en la Sección B.4.2 de tal forma que podamos tener una base para escoger uno de ellos y aplicarlo en nuestra versión multidimensional del algoritmo Scale Saliency. Por otra parte demostramos también que la complejidad temporal de nuestro método con respecto a la dimensionalidad de los datos pasa de ser exponencial a lineal. Nuestros experimentos demuestran que en la práctica el tiempo de ejecución del algoritmo Scale Saliency basado en el algoritmo k-d partition es notablemente más bajo que el de aquel basado en KNNGs o MSTs.

En primer lugar calculamos experimentalmente el tiempo de ejecución medio por píxel de los algoritmos de construcción de MSTs y KNNGs explicados. Este experimento demuestra que no siempre el algoritmo con menor complejidad teórica es el más rápido en la práctica. Sea n el número de nodos, y m el número de aristas. Dado que construimos MSTs y KNNGs a partir de grafos totalmente conexos, en nuestro caso tendremos $m = n^2$. La complejidad teórica de los diferentes métodos es:

- La complejidad del algoritmo de Kruskal es $O(m \log m)$.

- La complejidad del algoritmo de Prim es $O(n \log n)$.
- La complejidad asintótica del algoritmo de Katriel es $O(m + n \log n)$, que puede ser considerada lineal en el caso en el que $m >> n$. La complejidad esperada es $mT + O(n \log n + \sqrt{mn})$, donde T es el tiempo necesario para comprobar si una arista debe ser eliminada o no.
- Tras construir un k-d tree en $O(n \log n)$, la complejidad del algoritmo de construcción de un KNNG es $O(\log n)$, es decir, el coste de buscar los k vecinos más cercanos de cada nodo.

Nuestro experimento se explica a continuación. En primer lugar se escogieron aleatoriamente 100 píxeles de cada imagen de la base de datos de *Bristol*[7], una base de datos libre consistente en 29 imágenes construidas a partir de 31 bandas de una resolución de 256×256 filtradas espectralmente. Es decir, cada píxel tiene 31 valores asociados (31 dimensiones). Establecimos $s_{min} = 5$ y calculamos el tiempo medio de ejecución por píxel del algoritmo Scale Saliency usando los rangos de valores $s_{max} \in [8, 21]$ y $d \in [1, 31]$. En el caso del algoritmo de Katriel utilizamos el código proporcionado por los autores del algoritmo[8]. Nuestra implementación del algoritmo de Kruskal se basó en una ordenación mediante Quicksort y en una estructura de datos de tipo Disjoint-Set [Cormen et al., 2001]. Nuestra implementación de Prim se basó en una estructura de datos de tipo Pairing Heap [Fredman et al., 1986]. Con respecto al algoritmo de construcción de KNNGs, usamos $k = 4$ tanto para el algoritmo semidinámico [Bentley, 1990] como para el algoritmo no dinámico. En el primer caso la estructura de datos utilizada en las escalas más altas se actualiza para poder ser reutilizada en las escalas más bajas, lo que acelera en gran medida el funcionamiento del algoritmo. En el segundo caso, y en el caso de todos los algoritmos de construcción de MSTs, se debe construir un nuevo grafo para cada píxel y cada escala. El valor $k = 4$ proporciona un equilibrio entre un bajo tiempo de ejecución y una alta calidad de estimación. Finalmente, es necesario indicar que antes de construir un MST se añade ruido (esto no es necesario en el caso de la construcción de KNNGs – ver la Sección B.4.2). Esto podría afectar a los resultados. El

[7] http://psy223.psy.bris.ac.uk/hyper/
[8] http://www.mpi-inf.mpg.de/∼sanders/dfg/

resultado del experimento se muestra en la Fig. 4.7. Finalmente decidimos no mostrar los resultados del algoritmo de Kruskal; su complejidad depende fuertemente de la densidad de aristas y nuestros grafos son totalmente conexos. Por lo tanto, su tiempo de ejecución era mucho más elevado que el del resto de algoritmos del experimento. Es un algoritmo que tan solo debería ser aplicado a grafos dispersos o con un número bajo de vértices.

En la parte superior de la Fig. 4.7 se muestran los resultados para todos los algoritmos, mientras que en la parte inferior se muestran los resultados para todos los algoritmos excepto la versión no dinámica del algoritmo de construcción de KNNGs. El más rápido de todos ellos es el KNNG semidinámico, que era lo que se esperaba, ya que este algoritmo permite reutilizar información entre escalas. Por su parte, el algoritmo de Katriel es más rápido que el algoritmo de Prim, pero sus tiempos son parecidos. En ambos casos la dimensionalidad de los datos no parece afectar tanto al tiempo de ejecución como el rango de escalas. A pesar de que en el caso de la construcción de KNNGs la dimensionalidad de los datos parece ser un factor importante, el rango de escalas sigue siendo lo que más afecta al tiempo de ejecución. Finalmente se escogió la construcción semidinámica de KNNGs para comparar el algoritmo Scale Saliency multidimensional basado en grafos entrópicos con el basado en el algoritmo k-d partition.

Nuestro siguiente experimento, además de comparar las dos soluciones comentadas al final del párrafo anterior, demuestra que nuestro algoritmo puede manejar datos de alta dimensionalidad. Para ello utilizamos también la base de datos de *Bristol*. Nuestro algoritmo puede procesar directamente estas imágenes, pero sin embargo tuvimos que modificar el código del algoritmo Scale Saliency proporcionado por Kadir y Brady[9] para que éste pudiera procesar imágenes formadas por más de 3 bandas. En la Fig. 4.9 mostramos los resultados del experimento, en el que se calculó el tiempo medio de ejecución del algoritmo Scale Saliency completo para las 29 imágenes de la base de datos, conforme se iba incrementando el número de bandas de 1 a 31. En el caso del algoritmo Scale Saliency original fue necesario cuantizar los histogramas, debido al incremento exponencial de la complejidad espacial (y temporal). Por otra parte, la estimación de entropía

[9] http://www.robots.ox.ac.uk/~timork/salscale.html

a partir de KNNGs o el algoritmo k-d partition no requiere una cuantización de intensidad de este tipo. En la fila superior de la figura se muestran los resultados usando un rango de escalas entre $s_{min} = 5$ y $s_{max} = 8$, mientras que en la fila inferior se muestran los resultados usando un rango de escalas entre $s_{min} = 5$ y $s_{max} = 20$. En ambos casos, la columna de la derecha muestra en más detalle los tiempos de la versión de Scale Saliency basada en k-d partition.

Tal como se podría esperar el tiempo de ejecución del algoritmo de Kadir y Brady crece exponencialmente, ya que su complejidad es exponencial con respecto a la dimensionalidad de los datos. Ese sin embargo no es el caso de nuestro método. La complejidad del algoritmo Scale Saliency basado en KNNGs semidinámicos es $O(kn + n \log n)$ [Bentley, 1990], mientras que la del basado en el algoritmo k-d partition es $O(n \log n)$ [Stowell and Plumbley, 2009]. En ambos casos el tiempo de ejecución se incrementa linealmente con el número de dimensiones. El tiempo de ejecución del algoritmo k-d partition es notablemente inferior y permite procesar imágenes de alta dimensionalidad en un tiempo razonable. En el caso del algoritmo Scale Saliency de Kadir y Brady, su aplicación a datos de alta dimensionalidad es imposible, debido a los requerimientos temporales y espaciales del algoritmo. Aunque la versión del algoritmo basada en KNNGs resuelve este problema, su tiempo de ejecución sigue siendo excesivo. Algo que se puede observar en la Fig. 4.9, donde los tiempos de ejecución son comparados para dos rangos de escalas diferentes, es que este parámetro afecta en gran medida a la versión del algoritmo Scale Saliency basada en KNNGs. La versión basada en el algoritmo k-d partition parece verse menos afectado por este parámetro, aunque el impacto del mismo es aun notable.

B.4.7 Validación de los estimadores

En esta sección comprobaremos el funcionamiento de los diferentes estimadores presentados. Compararemos en primer lugar la estimación de entropía de Leonenko *et al.* (Sección B.4.2) con la del algoritmo k-d partition (Sección B.4.3), usando para ello dos tipos de distribuciones: Gaussiana y uniforme. La distribución normal $N(\mu, \sigma 2)$ es la de mayor entropía

entre todas las distribuciones de valores reales con media μ y desviación típica σ [Cover and Thomas, 1991]. La entropía teórica de una distribución Gaussiana en $\mathcal{R}^d$ con una matriz de covarianza Σ se puede calcular como

$$H_G = \frac{1}{2}\log((2\pi e)^d|\Sigma|) \ . \tag{B.40}$$

Por otra parte, la distribución uniforme en el rango $[a, b]$ es la distribución de máxima entropía entre todas las distribuciones continuas cuyo dominio se encuentra en el rango $[a, b]$ [Cover and Thomas, 1991]. La entropía teórica de una distribución uniforme en el rango $[a, b]$ se puede calcular como

$$H_u = \frac{1}{b-1}, a \leq b \ . \tag{B.41}$$

En ambos casos medimos la diferencia media (tras 100 ejecuciones) entre la entropía teórica para un número creciente de dimensiones y un número de píxeles en $R_{\mathbf{x}}$ para las escalas entre $s_{min} = 3$ y $s_{max} = 30$ y la entropía estimada por medio del método de Leonenko y el algoritmo k-d partition. Los resultados se pueden ver en la Fig. 4.10. En la parte superior se muestra la diferencia en el caso de la distribución uniforme en el rango $[-3, 3]^d$ y en la parte inferior la diferencia en el caso de una distribución Gaussiana de media cero y $\Sigma = I$ usando k-d partition (KDP) y Leonenko para diferentes valores de vecindad (k). Como se esperaba en todos los casos la estimación mejora conforme se aumenta el número de muestras a partir del cual se realiza la estimación. También en todos los casos el incremento en la dimensionalidad de los datos disminuye la calidad de la estimación. Los experimentos demuestran que ninguno de los dos estimadores supera al otro en todos los casos. El estimador de Leonenko aproxima mejor la entropía teórica de una Gaussiana, mientras que el algoritmo k-d partition aproxima mejor la entropía teórica de la distribución uniforme. Otra conclusión que se extrajo de estos experimentos es que el estimador de Leonenko no requiere un valor alto del parámetro k; los mejores resultados se obtienen para $k = 2$.

A pesar de los resultados obtenidos, se ha de tener en cuenta que en el caso del algoritmo Scale Saliency no es necesaria una estimación exacta de la entropía. Con que la entropia en función de la intensidad siga la misma tendencia que en el caso de la entropía de Shannon es suficiente. Por ello realizamos otro experimento para comprobar la tendencia de la entropía

obtenida a partir del estimador de Leonenko y del algoritmo k-d partition. El experimento consistió en tomar N muestras $x \in [0, 255]^d$ a partir de una distribución Gaussiana (uniforme), siendo N el número de píxeles en $R_{\mathbf{x}}$ para la escala $s_{max} = 30$ durante el algoritmo Scale Saliency. A continuación estimamos la entropia usando los dos métodos a comparar conforme se hacia disminuir el número de muestras, eliminando en cada iteración la muestra más lejana al centro de masas de las muestras y tomando la media tras 100 ejecuciones. El resultado para diferentes valores de d se muestra en la Fig. 4.11 (Fig. 4.12).

Las dos figuras anteriores también muestran la evolución de la entropía de Shannon, cuando ésta es estimada a partir de histogramas, para el caso de bajas dimensionalidades ($d = 2$ y $d = 3$), usando histogramas de 256 elementos. En el caso de la distribución Gaussiana el algoritmo k-d partition aproxima mejor la tendencia de la estimación basada en histogramas, incluso en el caso de dimensiones más altas. A partir de $d = 3$, la estimación basada en KNNGs converge muy rápido conforme aumenta N; por lo tanto, este método tiene menor capacidad de discriminación. Para el caso de la distribución uniforme ambos estimadores alcanzan pronto una asíntota; sin embargo, la estimación a partir del algoritmo k-d partition sigue aproximando mejor la forma de la curva de estimación de entropía a partir de histogramas. Este experimento demuestra que aunque ambos estimadores podrían ser utilizados en nuestro algoritmo multidimensional, el algoritmo k-d partition obtiene mejores resultados de estimación.

A continuación procedimos a realizar un experimento con el fin de validar nuestra divergencia basada en k-d partition, comparando sus resultados con los del test de Friedman-Rafsky. Para ello aplicamos el proceso experimental propuesto por Neemuchwala en su tesis [Neemuchwala, 2004]: comparamos la divergencia para dos conjuntos de muestras obtenidas a partir de dos distribuciones Gaussianas, con la misma media y varianza inicial, conforme aumentamos la distancia entre los centros de las distribuciones hasta que la probabilidad de que se superpongan las muestras de ambas sea muy baja. En la Fig. 4.13 mostramos los resultados de este experimento, usando 100 muestras por distribución (en concreto comparamos la divergencia del test de Friedman-Rafsky, en rojo, con la obtenida mediante nuestro método, en azul, conforme aumenta el número de dimensiones). En el caso de ambos

estimadores la divergencia (eje y) aumenta con la distancia entre los centros de las Gaussianas (eje x). La divergencia obtenida por Friedman-Rafsky siempre se encuentra en el rango $[0.5, 1]$, mientras que el rango de valores de divergencia obtenido a partir de nuestro método basado en k-d partition depende de la dimensionalidad de los datos. A pesar de que la tendencia en todos los casos es similar a la del test de Friedman-Rafsky, la divergencia máxima va disminuyendo con la dimensionalidad de los datos.

Obtenemos dos conclusiones de estos experimentos. En primer lugar, aunque el algoritmo Scale Saliency basado en k-d partition permite disminuir notablemente el tiempo de ejecución y permite que este método pueda ser aplicado a problemas en los que el uso de histogramas o grafos entrópicos no sería factible (en la sección anterior hemos llegado a mostrar resultados para hasta 31 dimensiones), la estimación se va degradando conforme aumenta la dimensionalidad de los datos. Por lo tanto, la aplicación de nuestro método estará restringida a problemas en los que ésta no sea muy elevada. Esta restricción no es debida a nuestro algoritmo, sino que a la naturaleza del propio método de k-d partition, el cual requiere al menos 2^d muestras para obtener una estimación consistente de la entropía. Este requerimento no se cumple durante el algoritmo Scale Saliency para el caso de las dimensionalidades más altas, en las cuales el número de muestras suele encontrarse entre menos de 10 (para $s = 3$) y unas 3000 (para $s = 30$). En segundo lugar, para dimensionalidades más bajas, el rango de posibles valores de divergencia para el método basado en k-d partition es más amplio que en el caso de utilizar el test de Friedman-Rafsky, para el que la divergencia nunca es menor de 0,5. Por lo tanto, nuestra medida de divergencia es más consistente en el caso de dimensionalidades bajas.

B.4.8 Calidad de las características extraídas

En esta sección comparamos la calidad de las características extraídas mediante el algoritmo de Kadir y Brady y mediante nuestro algoritmo multidimensional, tanto en el caso de grafos entrópicos como en el caso de aplicar k-d partition. Para ello utilizaremos la prueba de repetibilidad y las secuencias de imágenes de Mikolajczyk *et al.* [Mikolajczyk et al., 2005b], ya que tanto el código de la prueba como las secuencias de imágenes se pueden

descargar libremente de Internet[10] y además es la prueba normalmente utilizada en la literatura para comparar la calidad de las características extraídas por diferentes algoritmos.

Cada secuencia de imágenes se compone de una imagen de referencia y cinco imágenes transformadas, obtenidas tras aplicar el mismo tipo de transformación a la primera imagen en orden creciente de magnitud. Las secuencias de imágenes son: *graph* (cambio de punto de vista en una escena estructurada), *wall* (cambio de punto de vista en una escena con texturas), *boat* (cambio de escala en una escena estructurada), *bark* (cambio de escala en una escena con texturas), *bikes* (emborronamiento en una escena estructurada), *trees* (emborronamiento en una escena con texturas), *ubc* (compresión JPEG) y *leuven* (variación de la iluminación de la escena). Todas estas secuencias están formadas por imágenes en color, excepto la secuencia *boat*. que no usaremos en nuestro experimento de repetibilidad.

El experimento consistió en lo siguiente: en primer lugar se aplicaron todos los algoritmos de extracción de características a comparar a las imágenes de todas las secuencias. A continuación, para cada algoritmo y cada secuencia, se calculó la repetibilidad entre la imagen de referencia y el resto de imágenes de dicha secuencia. La **repetibilidad** se define como la cantidad relativa de solapamiento entre las características detectadas en la imagen transformada y la imagen de referencia, tras proyectar las segundas en la primera por medio de una homografía. Se considera que dos características se solapan si el siguiente error se encuentra por debajo de un determinado umbral ϵ:

$$1 - \frac{R_{\mu_i} \bigcap R_{(H^T \mu_1 H)}}{(R_{\mu_i} \bigcup R_{(H^T \mu_1 H)})} < \epsilon \ , \tag{B.42}$$

donde R_μ es la región definida por los parámetros μ de la elipse (en este caso del círculo) que representa la característica extraída, y H es la homografía que relaciona la imagen de referencia con la imagen transformada. Por lo tanto, la repetibilidad es la proporción entre el número de características solapadas y el mínimo número de características detectadas en ambas imágenes. Tan solo se consideran en la medida aquellas características que sean detectadas

[10] http://www.robots.ox.ac.uk/∼vgg/research/affine

en lugares de las dos imágenes que correspondan a partes comunes de la escena.

El resultado de este experimento es una gráfica para cada secuencia de imágenes que muestra cómo afecta a cada algoritmo el ir agudizando la transformación aplicada a la imagen de referencia. Es de esperar que la repetibilidad disminuya conforme la magnitud de la transformación aumente. Los resultados del experimento se muestran en la Fig. 4.15. Los algoritmos comparados fueron los siguientes: el algoritmo Scale Saliency de Kadir y Brady empleando tan solo tonos de gris (*KB1D*), el algoritmo Scale Saliency de Kadir Brady utilizando información de color (*KB3D*), nuestro algoritmo multidimensional basado en el estimador de Leonenko y en la divergencia de Friedman-Rafsky (ambos a partir de KNNGs) usando información de color (*Leo*), nuestro algoritmo multidimensional basado en k-d partition para la estimación de la entropía y de la divergencia usando información de color (*KDP*), y el método *KDP* usando información de color pero cambiando la forma en la que las dimensiones se van seleccionando durante la partición del espacio (*KDPv*). En el algoritmo k-d partition de Stowell y Plumbley las dimensiones son seleccionadas secuencialmente, mientras que en nuestro algoritmo propuesto *KDPv* se selecciona en cada llamada recursiva la dimensión para la que la varianza sea máxima. En todos los casos se extrajo para cada imagen el 1% de características más salientes.

En general la información de color mejora la repetibilidad de las características extraídas. Esto es así porque ésta aumenta la distinguibilidad. La única excepción se produce en el caso de la secuencia *bark*. Los resultados del algoritmo de Kadir y Brady usando color son siempre mejores que los de nuestra solución multidimensional. Esto es así por diferentes motivos. En el caso del algoritmo *Leo* pequeñas variaciones en la posición de un nodo pueden suponer la construcción de un KNNG diferente, afectando a la calidad de la estimación de la entropía. Por otra parte, en el caso del algoritmo k-d partition, las celdas tienen volumen finito, lo que degrada los resultados de la estimación [Stowell and Plumbley, 2009].

Si comparamos los algoritmos *Leo* y *KDP* veremos que cada uno de ellos se comporta mejor para determinados tipos de transformaciones, de tal forma que no podemos considerar a ninguno de los dos mejor que el otro en cualquier circunstancia. Este resultado, junto a los tiempos de ejecución

obtenidos durante los experimentos de la Sección B.4.6, nos hicieron decantarnos por el algoritmo *KDP* como base de nuestro algoritmo Scale Saliency multidimensional (Sección B.4.4). Curiosamente la repetibilidad del método *KDPv* siempre fue menor que la del método *KDP*. Parece ser que en el caso de baja dimensionalidad de los datos (en nuestro experimentos de repetibilidad hemos utilizado tres dimensiones) el particionado de los datos siguiendo un orden secuencial produce mejores resultados. Los peores resultados para nuestra versión multidimensional del algoritmo Scale Saliency se obtienen en el caso de la secuencia *ubc*. Los estimadores parecen ser muy sensibles a la presencia de regiones homogéneas en la imagen.

B.4.9 El efecto de la dimensionalidad de los datos en el número de características extraídas

El número total de características salientes detectadas puede afectar a la calidad y repetibilidad de los resultados obtenidos por un algoritmo extractor de características [Mikolajczyk et al., 2005b]. Diferentes algoritmos pueden detectar un número diferente de características en una misma imagen debido a que estos algoritmos se centran en diferentes propiedades de la misma. En el caso del algoritmo Scale Saliency multidimensional llevamos a cabo un experimento para comprobar si este factor se veía afectado por el tipo de estimador de entropía y divergencia utilizado (grafos entrópicos, k-d partition, histogramas). También deseábamos comprobar el efecto que tendría en el número de características salientes detectadas la dimensión de los datos de entrada.

En el algoritmo Scale Saliency es un parámetro el que determina el número final de características salientes detectadas. El paso final del algoritmo era seleccionar un porcentaje de todos los picos de entropía encontrados en el espacio de escalas, tras ponderarlos, y en orden de mayor a menor saliencia. Sin embargo, este parámetro (el porcentaje de características salientes) no permite seleccionar una cantidad exacta de características a extraer, debido al paso de supresión de no máximos. Por lo tanto, en nuestros experimentos utilizaremos el número de picos de entropía detectados en el espacio de escalas como un indicador de la cantidad de características salientes detectadas, en lugar del número final de características.

En la Fig. 4.16 se pueden ver los primeros resultados de nuestro experimento. En este experimento calculamos el número medio de picos de entropía encontrados por la versión multidimensional del algoritmo Scale Saliency para las imágenes de la base de datos de *Bristol* conforme aumentamos el número de dimensiones (el número de bandas de las imágenes de entrada utilizados para estimar la entropía). En la gráfica comparamos los resultados del algoritmo basado en la estimación de Leonenko y Friedman-Rafsky con el algoritmo basado en la estimación a partir del método k-d partition. Cuando la dimensionalidad de los datos es baja (menos de 5 dimensiones), el Scale Saliency basado en k-d partition obtiene muchos más picos de entropía. Para dimensionalidades mayores los resultados son muy similares, aunque el número de picos de entropía detectados por el algoritmo k-d partition sigue siendo ligeramente mayor. Por lo tanto, podríamos considerar a ambos algoritmos equivalentes en cuanto al número de características extraídas y podrían ser utilizados indistintamente si este fuera el único factor relevante.

En la Fig. 4.17 comparamos los resultados obtenidos usando estimación a partir de grafos entrópicos, k-d partition e histogramas (tal cual se hace en el algoritmo Scale Saliency de Kadir y Brady). En este caso no se consideró factible aplicar el algoritmo basado en histogramas a datos de más de cuatro dimensiones (más de cuatro capas en las imágenes del conjunto de *Bristol*), así que tan solo podemos comparar todos estos algoritmos parcialmente. El número de picos de entropía en este rango (de 1 a 4 dimensiones) es notablemente mayor en el caso de la estimación basada en histogramas. Este hecho podría ser la causa de que el algoritmo de Kadir y Brady obtuviera mejores resultados en los experimentos de repetibilidad de la sección anterior, ya que como se ha comentado anteriormente, el número de características detectadas puede afectar a los resultados de repetibilidad.

Otro dato a tener en cuenta es que conforme el número de dimensiones aumenta el número de picos detectados disminuye. Uno de los experimentos que llevamos a cabo fue el de aplicar nuestro algoritmo Scale Saliency multidimensional a imágenes con información en 128D, en las cuales se calculó un descriptor SIFT [Lowe, 1999] para cada pixel de la imagen, usando para ello una escala fija. En la mayoría de los casos el algoritmo no detectó ningún pico de entropía, o lo que es lo mismo, el algoritmo no extrajo ninguna

característica saliente. La conclusión aquí es la misma que en la Sección B.4.7: la dimensionalidad de los datos de entrada para la que nuestro algoritmo puede ser aplicado está limitada.

B.4.10 Ejemplo de aplicación: categorización de texturas

En esta sección aplicaremos nuestro algoritmo al problema de la categorización de texturas. Dada una imagen conteniendo una única textura, el objetivo de un algoritmo de categorización de texturas es indicar a que categoría pertenece dicha imagen. Nuestros experimentos están basados en la representación de texturas de Lazebnik *et al.* [Lazebnik et al., 2005], que propusieron representar cada textura como una firma formada a partir de un conjunto de características extraídas de la imagen. Dichas características eran extraídas a partir de la intensidad en tonos de gris. Nuestra hipótesis es que incluir información adicional en el proceso de extracción de características mejorará la calidad de la categorización.

Veamos en primer lugar como Lazebnik *et al.* construyen una firma para una determinada imagen. En primer lugar, se extraen características de la imagen haciendo uso del algoritmo Harris afín [Mikolajczyk and Schmid, 2004b] y el detector de Gårding y Lindeberg [Gårding and Lindeberg, 1996] (ver Sección B.2). El hecho de combinar el resultado de ambos algoritmos, que se centran en la extracción de características de diferente naturaleza, hace que se obtengan mejores resultados. Las regiones extraídas se representan como elipses. Las elipses son normalizadas, pasando a ser círculos unitarios, y se calcula un descriptor para cada una de ellas. Para ello utilizan dos descriptores invariantes a rotación (ya que Lazebnik *et al.* afirman que la estimación de la orientación suele estar sujeta a errores): *spin images* y *Rotation Invariant Feature Transform* (RIFT).

Un spin image es un histograma 2D que codifica la distribución de intensidades en la vecindad de un determinado píxel (el centro de la característica, en este caso). Las dos dimensiones del histograma son la intensidad I y la distancia del centro d. En este histograma cada píxel contribuye tanto a su posición correspondiente como a las posiciones vecinas. Dado un pixel $\mathbf{x}$, su contribución a la posición del histograma (d, i) viene

dada por

$$exp\left(-\frac{(|\mathbf{x}-\mathbf{x}_0|-d)^2}{2\alpha^2}-\frac{|I(\mathbf{x})-i|^2}{2\beta^2}\right) \quad , \tag{B.43}$$

donde $\mathbf{x}_0$ es el centro de la característica detectada, $I(\mathbf{x})$ es la intensidad del píxel $\mathbf{x}$, y α y β son dos parámetros que especifican cuánto afecta cada píxel a las posiciones vecinas del histograma. El descriptor RIFT es una generalización invariante a rotación del descriptor SIFT de Lowe [Lowe, 1999]. El círculo unitario es dividido en anillos concéntricos de igual anchura. El resultado es un histograma 2D, en el que una dimensión representa la orientación θ de un píxel relativa a la orientación del píxel central y la otra representa la distancia al centro d (el anillo).

El siguiente paso es construir la firma de la imagen. Se aplica un algoritmo de agrupamiento para los descriptores de dicha imagen, de manera independiente (es decir, no se aplica el agrupamiento a un conjunto de imágenes para construir un *vocabulario visual*, como en el caso de los algoritmos de Bag of Words [Sivic and Zisserman, 2003]). Se emplea un simple agrupamiento aglomerativo: a partir de un estado inicial, en el que habrá tantos grupos como descriptores, el algoritmo iterativamente va uniendo el par de grupos cuyas medias estén más cercanas, hasta alcanzar un determinado umbral de distancias. Tras el agrupamiento, la imagen se representa mediante un vector $\{(c_1, w_1), (c_2, w_2), \ldots, (c_k, w_k)\}$ en el que k es el número de grupos, c_i es el centro del grupo i y w_i es el peso relativo del grupo i (es decir, el número de descriptores pertenecientes al grupo dividido entre el número total de descriptores extraídos de la imagen). Mediante esta representación es posible comparar texturas por medio de la *Earth Mover's Distance* (EMD) [Rubner et al., 2000]. La EMD entre dos firmas $S_1 = \{(c_1, w_1), (c_2, w_2), \ldots, (c_k, w_k)\}$ y $S_2 = \{(d_1, u_1), (d_2, u_2), \ldots, (d_k, u_k)\}$ se calcula como

$$d(S_1, S_2) = \frac{\sum_i \sum_j f_{ij} d(c_i, d_j)}{\sum_i \sum_j f_{ij}} \quad , \tag{B.44}$$

donde los escalares f_{ij} son valores de flujo que se obtienen resolviendo un problema de programación lineal, y $s(c_i, d_j)$ es la distancia euclídea entre los centros de los grupos c_i y d_j. Estos centros son o bien spin images o bien

descriptores RIFT.

Como se ha indicado anteriormente, Lazebnik *et al.* extraen características a partir de los tonos de gris de la imagen. Nuestra propuesta es aplicar nuestro algoritmo Scale Saliency multidimensional para obtener características salientes a partir de información multidimensional. Para ello escogimos aplicar un banco de filtros de Gabor para obtener un descriptor multidimensional para cada píxel de la imagen, ya que los filtros de Gabor han sido ampliamente utilizados en el campo del análisis de texturas [Bianconi and Fernández, 2007]. Un filtro de Gabor 2D se define como una función harmónica multiplicada por una función Gaussiana

$$G(x,y) = exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \quad , \tag{B.45}$$

donde $x' = xcos\theta + ysin\theta$ y $y' = -xsin\theta + ycos\theta$, siendo λ la longitud de onda del filtro, θ su orientación, ψ la fase, σ la varianza de la Gaussiana, y γ un ratio de aspecto que permite que el filtro tenga forma de elipse. Nuestro banco de filtros consistió en 15 filtros de Gabor que fueron aplicados a cada píxel de la imagen, cada uno de ellos con diferentes orientaciones y longitudes de onda. Estos filtros se muestran en la Fig. 4.18. Todos los filtros comparten el mismo valor de σ, lo cual quiere decir que todos tendrán la misma escala, ya que obtuvimos mejores resultados de esta forma que utilizando filtros de escalas diferentes en el banco de filtros. Tras aplicar dicho banco de filtros a la imagen, utilizamos la versión multidimensional del algoritmo Scale Saliency para obtener características a partir de información 15D. La firma de cada textura se extrae siguiendo el método explicado anteriormente.

Para evaluar la calidad de la representación multidimensional de las texturas llevamos a cabo un proceso de recuperación de imágenes, utilizando el conjunto de imágenes de texturas de *Brodatz*[11]. Este conjunto de imágenes contiene 111 categorías, conteniendo a su vez cada una 9 imágenes en tonos de gris, con lo que se tiene un total de 999 imágenes. Todas las imágenes muestran una única textura. Es un conjunto de datos bastante complicado, ya que es difícil distinguir entre las imágenes pertenecientes a algunas categorías, incluso en el caso de un observador humano. Algunos ejemplos de texturas se muestran en la Fig. 4.19.

[11] http://www.ux.uis.no/~tranden/brodatz.html

En el experimento de recuperación de imágenes utilizamos todas las imágenes del conjunto de datos una vez como imagen de entrada. Para cada imagen de entrada seleccionamos las imágenes del conjunto total de *Brodatz* más parecidas en orden descendiente de EMD. El resultado es una gráfica denominada *average recall*: conforme aumentamos el número de imágenes obtenidas del conjunto de datos para la imagen de entrada, mostramos el numero medio de dichas imágenes que pertenecen a su misma categoría, dividido entre el número total de imágenes de su categoría. Con tal de evitar el posible efecto de obtener un número diferente de características para cada imagen (debemos recordar que este parámetro afecta en gran medida a la calidad de las características extraídas; ver Sección B.4.9 y [Mikolajczyk et al., 2005b]), tanto el algoritmo Scale Saliency de Kadir Brady como la versión multidimensional del mismo fueron modificados para devolver siempre 150 características *tras la supresión de no mínimos*.

Mostramos los resultados del experimento en la Fig. 4.20. Comparamos los siguientes métodos: algoritmo Scale Saliency de Kadir y Brady con descriptores RIFT (*kadirrift*), algoritmo Scale Saliency de Kadir y Brady con spin images (*kadirspin*), algoritmo Scale Saliency de Kadir y Brady combinando RIFT y spin images (*kadir*), Scale Saliency multidimensional con RIFT (*kdpeerift*), Scale Saliency multidimensional con spin images (*kdpeespin*) y Scale Saliency multidimensional combinando RIFT y spin images (*kdpee*). Para combinar los descriptores RIFT y spin images, la distancia total entre dos imágenes se calcula sumando las EMDs normalizadas para cada descriptor individual. Como puede verse en la gráfica, utilizar información multidimensional mejora los resultados en el caso de cualquier descriptor. En la Fig.4.21 mostramos también algunos ejemplos de resultados de aplicación tanto del algoritmo Scale Saliency de Kadir y Brady como de nuestra versión multidimensional a algunas imágenes del conjunto de datos de Brodatz.

A pesar de obtener mejores resultados con información multidimensional, su impacto no es tan notable como el que se podría producir por la elección de un descriptor adecuado. Como se puede ver en la Fig. 4.20, los peores resultados se obtienen en el caso de utilizar exclusivamente spin images, mejorándose si se aplica exclusivamente el descriptor RIFT. La mejora más significativa se produce al combinar ambos. Utilizar información multidimensional tan solo mejora estos resultados

ligeramente. En nuestra opinión estos resultados podrían ser mejorados muy notablemente seleccionando un banco de filtros de Gabor óptimo. Este es un problema combinatorio que podría ser resuelto mediante técnicas de Aprendizaje Computacional, como algoritmos de selección de características. Esto queda fuera de los objetivos de esta tesis y se ha dejado como trabajo futuro.

B.4.11 Conclusiones

En esta sección hemos presentado dos posibles soluciones para la generalización del algoritmo Scale Saliency al dominio multidimensional. La primera se basa en la estimación de entropía y auto-disimilitud por medio de Árboles de Longitud Mínima y Grafos de K-Vecinos Más Cercanos. La segunda se apoya en la estimación de entropía y auto-disimilitud por medio del algoritmo k-d partition. Evaluamos experimentalmente ambas soluciones y aplicamos nuestro algoritmo multidimensional a la categorización de texturas. Las principales contribuciones de esta sección fueron:

- Evaluamos la aplicación de métodos alternativos de estimación de entropía y auto-disimilitud a partir de grafos al algoritmo Scale Saliency: estimación de la entropía de Shannon a partir de la α-entropía de Rényi (las máximas desventajas de este método son su inestabilidad y que es un método especialmente diseñado para distribuciones Gaussianas), estimación de la entropía siguiendo el método de Kozachenko y Leonenko, y el algoritmo de Friedman-Rafsky que proporciona una aproximación de la divergencia de Henze-Penrose.

- Estudiamos la aplicación del algoritmo k-d partition, un estimador rápido de entropía basado en la partición de los datos, a la generalización multidimensional del método Scale Saliency.

- Introducimos una nueva medida de divergencia basada en el algoritmo k-d partition y en la distancia de variación total que permite estimar la auto-similitud entre escalas. Nuestros experimentos demostraron que los resultados obtenidos con nuestra nueva divergencia son comparables a los del método de Friedman-Rafsky. Además, la

distinguibilidad de nuestra divergencia es mayor en el caso de no muy altas dimensionalidades.

- Analizamos experimentalmente todos los métodos de estimación presentados en la sección en términos de eficiencia computacional, calidad de estimación y calidad y cantidad de características extraídas. En primer lugar demostramos que nuestro algoritmo de Scale Saliency multidimensional reduce la complejidad con respecto a la dimensionalidad de los datos de exponencial a linear. El método basado en k-d partition es notablemente más eficiente que el basado en gafos. Ambos métodos, sin embargo, son equivalentes en cuanto a la calidad de la estimación obtenida y de las características extraídas. Finalmente demostramos que ambas soluciones producen como resultado un número similar de características salientes extraídas en el caso en el que la dimensionalidad de los datos sea superior a 5. Para dimensionalidades más bajas el método basado en k-d partition proporciona muchas más características salientes. Sin embargo, si comparamos ambos métodos con el Scale Saliency de Kadir y Brady comprobamos que éste último permite extraer una cantidad mucho mayor de características.

- Presentamos una aplicación práctica del algoritmo Scale Saliency multidimensional. Demostramos que los datos multidimensionales pueden mejorar los resultados de la tarea de categorización de texturas en el caso de utilizar el método de representación de texturas de Lazebnik *et al.* [Lazebnik et al., 2005].

Hemos proporcionado resultados experimentales con datos de hasta 31 dimensiones. El algoritmo original de Kadir y Brady no puede tratar datos de una dimensionalidad tan alta. Usando nuestro método multidimensional basado en k-d partition somos capaces de procesar una imagen 31D con una resolución de 256x256 en menos de cuatro minutos en un ordenador de sobremesa normal. Sin embargo, el rango de dimensiones en el que podemos aplicar nuestro algoritmo multidimensional sigue estando limitado. Conforme incrementamos este factor la divergencia basada en k-d partition produce peores resultados y el número de características extraídas

disminuye.

Nuestro trabajo futuro incluye el desarrollo de nuevas aplicaciones para las que el algoritmo Scale Saliency multidimensional sea útil. Ejemplos de posibles aplicaciones son: análisis de vídeo (cada fotograma en la secuencia de vídeo equivaldría a una banda en una imagen multidimensional), categorización de imágenes (¿Cómo afecta el uso de información adicional, como la salida proporcionada por un banco de filtros, o información sobre magnitud y orientación de gradiente, a los resultados obtenidos con este tipo de aplicaciones?) y por supuesto el análisis de imágenes hiperespectrales. En el campo de la categorización de texturas deberíamos estudiar el resultado de aplicar diferentes bancos de filtros de Gabor, e incluso el resultado de utilizar diferentes datos de entrada. Como se mencionó anteriormente, esto es un problema combinatorio que podría ser tratado con técnicas de Aprendizaje Computacional como por ejemplo selección de características.

B.5 Conclusiones generales

En esta última sección presentamos un resumen detallado de las aportaciones de esta tesis. También destacamos las posibles limitaciones de los métodos y algoritmos presentados. Finalmente hablamos de diferentes posibilidades de cara a orientar nuestro trabajo futuro.

B.5.1 Resumen de la tesis

En esta tesis se ha trabajado dentro del marco de la extracción de características en imágenes. Los algoritmos de extracción de características detectan regiones de alta distinguibilidad. Estas regiones de alta distinguibilidad constituyen la entrada para otros algoritmos de visión de alto nivel, como algoritmos de reconocimiento de objetos, localización robótica, etc. Dada su importancia, en esta tesis se presenta un estudio en profundidad de este tipo de algoritmos, desde los más simples detectores de puntos esquina de los setenta (aunque también se presentan algoritmos recientes dentro de esta categoría) hasta el estado del arte actual en extractores de características, en el que podemos encontrar algoritmos que extraen regiones salientes con invarianza a transformaciones afines. Uno de

los puntos de este estudio al que se le ha prestado más atención es al de la descripción de la representación del espacio de escalas.

En el estudio se ha incluido una descripción detallada del algoritmo Scale Saliency de Kadir y Brady, que es el objeto de investigación de esta tesis. Es un algoritmo basado en la Teoría de la Información (hace uso de la entropía de Shannon) para detectar regiones de saliencia local en la imagen, esto es, regiones que son local y altamente salientes. Es un algoritmo interesante por dos motivos: el hecho de que se base en Teoría de la Información es consistente con el objetivo de encontrar regiones altamente informativas, y además se trata de un algoritmo cuya alta eficacia para tareas de categorización de imágenes ha sido demostrado previamente en la literatura. Sin embargo, el algoritmo Scale Saliency sufre varias limitaciones. La más notable de ellas es que se trata del algoritmo más lento entre todos los del estado del arte. Además, su complejidad computacional aumenta exponencialmente con el número de dimensiones de los datos a procesar. Las dos principales aportaciones de la presente tesis van dirigidas a mejorar estos aspectos del algoritmo.

Nuestra primera contribución principal consiste en un algoritmo de filtrado Bayesiano que es capaz de descartar puntos no interesantes de una imagen antes de aplicarle el algoritmo Scale Saliency. La consecuencia de esto es que el tiempo total de aplicación del algoritmo es notablemente menor. Nuestra hipótesis inicial fue que si una región era homogénea o no saliente en las escalas más altas, con mucha probabilidad también lo sería en las escalas más bajas. Tras un análisis empírico de la evolución de la función de entropía en el espacio de escalas realizamos un análisis estadístico en mayor profundidad que demostró que la entropía de un punto en la escala máxima podría ser considerada como un límite superior de su entropía a lo largo del rango de escalas. A partir de este resultados diseñamos un primer algoritmo de filtrado que funcionaba de la siguiente manera: dado un umbral de entropía, todos aquellos puntos de la imagen cuya entropía en la máxima escala estuviera por debajo de dicho umbral debían ser descartados antes de aplicar el algoritmo Scale Saliency al resto de la imagen.

La mayor limitación de esta solución simple radica en la dificultad de escoger un umbral adecuado que pueda ser aplicado a un conjunto de imágenes, en lugar de a una sola. Por ello, proponemos un algoritmo

de aprendizaje que permite aprender un umbral de entropía para una determinada categoría de imágenes. Este algoritmo está basado en la Teoría de la Información y en el método de detección de aristas de Konishi *et al*. En primer lugar, dado un conjunto de imágenes de entrenamiento pertenecientes a una misma categoría, estimamos dos distribuciones de probabilidad que indican la probabilidad de que un píxel de una imagen pertenezca o no las regiones más salientes de la misma dado su valor de entropía a escala máxima (siendo este valor de entropía relativo al máximo valor en la imagen a escala máxima). A continuación, se mide el valor de la Información de Chernoff entre estas dos distribuciones para comprobar si son lo suficientemente separables como para poder aprender un umbral válido. Después, calculamos la divergencia de Kullback-Leibler entre estas distribuciones, lo que nos da un rango de umbrales válidos para dicha categoría de imágenes. Si se selecciona un valor bajo dentro de este rango, la probabilidad de descartar regiones salientes es baja, pero la cantidad de puntos descartados también lo es. Por otra parte, tomando el máximo umbral posible hace que la aplicación del algoritmo Scale Saliency sea más rápida, pero también existe una mayor probabilidad de que se descarten puntos que sí que forman parte de las regiones más salientes de la imagen. Es interesante también recalcar el hecho de que la anchura de este rango de umbrales y el valor obtenido de la Información de Chernoff están relacionados.

Para demostrar que nuestro método produce una gran disminución del coste computacional del algoritmo Scale Saliency, sin afectar en gran medida al resultado final, realizamos una serie de experimentos utilizando dos bases de datos diferentes de imágenes. También estudiamos cómo afectan los distintos parámetros del algoritmo, como por ejemplo el número de imágenes de entrenamiento utilizadas, el número de regiones salientes a extraer, o el rango de escalas, al resultado final. Además, mostramos una aplicación práctica de nuestro algoritmo en el campo de la localización robótica.

Nuestra segunda aportación principal en esta tesis es la aplicación de métodos alternativos de estimación de entropía y divergencia con el objetivo de disminuir el orden computacional, con respecto al número de dimensiones de los datos procesados, del algoritmo Scale Saliency. Se estudiaron dos tipos de métodos: métodos basados en grafos (Árboles de Longitud Mínima y Grafos de K-Vecinos Más Cercanos) y el algoritmo k-d partition de

Stowell y Plumbley. En el caso de la solución basada en k-d partition diseñamos un nuevo método de estimación de divergencia inspirado por la distancia de variación total. Nuestros experimentos con una base de datos de imágenes hiperespectrales (con un total de 31 bandas, es decir, 31 valores por pixel) demostraron que cuando aplicamos estas técnicas conjuntamente con el algoritmo Scale Saliency se consigue una complejidad lineal en lugar de exponencial con respecto al número de dimensiones. Nuestros experimentos también demostraron que los métodos basados en grafos son más lentos, mientras que el algoritmo k-d partition nos permite construir una versión multidimensional del algoritmo Scale Saliency que es factible. Nuestros algoritmos también probaron la validez de las diferentes medidas de estimación estudiadas. Nuestro experimento de cálculo de la repetibilidad midió la calidad de las características extraídas. El rendimiento de nuestra solución multidimensional en el caso de imágenes a color (no se pudo aplicar el algoritmo Scale Saliency a datos de mayor dimensionalidad) es menor que el obtenido con el algoritmo Scale Saliency original, pero es mayor que en el caso de utilizar este último para extraer regiones de interés a partir de intensidades en escala de grises.

Por último, aplicamos nuestro algoritmo de Scale Saliency multidimensional al problema de la categorización de texturas. Basándonos en el trabajo de Lazebnik *et al.*, describimos cualquier imagen de la base de datos de texturas de *Brodatz* a partir de un conjunto de características extraídas usando intensidades en tonos de gris y también usando información 15D obtenida tras aplicar un banco de filtros de Gabor a todos los píxeles de la imagen. Nuestros experimentos demostraron que la información multidimensional aumentó el rendimiento de la categorización.

B.5.2 Trabajo futuro

Las aportaciones presentadas en esta tesis mejoran en gran medida el rendimiento del algoritmo Scale Saliency de Kadir y Brady. Sin embargo, también hemos identificado algunas limitaciones en nuestros métodos que podrían ser objeto de posteriores análisis, así como posibles nuevas ramas de investigación que podrían surgir a partir de nuestro trabajo. En primer lugar, sería interesante analizar en mayor profundidad el papel de la Información

de Chernoff en nuestro método de filtrado Bayesiano. A pesar de que afirmamos que parece existir una relación entre el valor de esta medida y el rango de umbrales válidos de entropía para una determinada categoría de imágenes, no hicimos un análisis en profundidad de esta relación.

El objetivo de la Información de Chernoff es medir la *homogeneidad* de una categoría de imágenes, de tal forma que podamos determinar si es lo suficientemente homogénea como para poder aprender un umbral válido para ella. Este concepto es similar al de variabilidad intraclase, una medida aplicada en algoritmos de agrupamiento junto a la variabilidad interclase para obtener una buena partición de los datos. En nuestros experimentos todas las imágenes fueron etiquetadas manualmente (es decir, asignadas manualmente a una determinada categoría). Esta categorización siguiendo un criterio humano podría no ser óptima. Un agrupamiento no supervisado de las imágenes podría proporcionar una separación en categorías diferente que mejorara el rendimiento del algoritmo de filtrado. En este caso, la Información de Chernoff podría actuar como una medida de variabilidad intraclase.

Por otra parte, nuestro filtrado Bayesiano no está diseñado a partir de la versión con invarianza afín del algoritmo Scale Saliency. La invarianza afín se obtiene mediante la extracción de regiones anisotrópicas (elipses) en lugar de regiones isotrópicas (círculos); gracias a la invarianza afín se consigue una mejora en la estabilidad de las características extraídas. La versión con invarianza afín del algoritmo Scale Saliency introduce dos parámetros nuevos, además de la escala. Podríamos ampliar nuestro análisis de la evolución de la entropía para incluirlos. También sería deseable saber si nuestro método podría ser aplicado a otros algoritmos de extracción de características, como la versión afín de Harris, por ejemplo.

Creemos que se podría mejorar la eficiencia temporal del algoritmo Scale Saliency implementando filtros adicionales. Estos filtros podrían ser dispuestos en forma de cascada, de manera que la salida de un filtro en un determinado nivel de la cascada pasara a ser la entrada del filtro en el nivel inmediatamente inferior. Así podríamos aplicar los filtros de mayor complejidad computacional en las últimas etapas, a las que llegarían menos puntos para ser procesados. La idea de usar una cascada de filtros ha sido aplicada con éxito anteriormente. Dos ejemplos de ello son el algoritmo de

detección de caras de Viola y Jones y el método de detección de textos de Yuille *et al*.

Nos centramos ahora en nuestra versión multidimensional del algoritmo Scale Saliency. Esta aportación permite la aplicación de dicho algoritmo a un nuevo conjunto de problemas. Sin embargo, el número de dimensiones a utilizar sigue estando limitado, ya que la divergencia basada en el algoritmo k-d partition tiende a degradarse y el número de características detectadas tiende a disminuir conforme aumenta la dimensionalidad. Este límite debería ser estudiado. Además podríamos plantearnos una pregunta interesante: ¿por qué el número de picos de entropía aumenta con la dimensionalidad en el caso de utilizar el algoritmo Scale Saliency original, pero disminuye en el caso de utilizar la versión basada en grafos o en el algoritmo k-d partition?

Finalmente, obtuvimos resultados prometedores en nuestros experimentos de categorización de texturas. Estos resultados podrían mejorarse diseñando un método que permitiera escoger el banco de filtros de Gabor más adecuado para esta tarea. La conclusión de este experimento es que las regiones de interés extraídas a partir de datos multidimensionales parecen mejorar el rendimiento de los algoritmos de visión que se basan en ellas. Podríamos pensar en diferentes aplicaciones para las que la extracción de características a partir de información multidimensional podría ser útil: análisis de vídeo, procesamiento de imágenes hiperespectrales, etc.

Bibliography

[Aguilar, 2006] Aguilar, W. (2006). Reconocimiento de objetos basado en la correspondencia estructural de características locales. Master's thesis, Universidad Autónoma de México.

[Arkin and Balch, 1998] Arkin, R. C. and Balch, T. (1998). *AI-based mobile robots: case studies of succesful robot systems*. MIT Press.

[Asada and Brady, 1986] Asada, H. and Brady, M. (1986). The curvature primal sketch. *IEEE Transactions on Pattern Analisis and Machine Intelligence*, 8:2–14.

[Babaud et al., 1986] Babaud, J., Witkin, A. P., Baudin, M., and Duda, R. O. (1986). Uniqueness of the gaussian kernel for scale space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):26–33.

[Beardsley et al., 1996] Beardsley, P., Torr, P., and Zisserman, A. (1996). 3d model acquisition from extended image sequences. In *Proceedings of the 4th European Conference on Computer Vision*, volume 2, pages 683–695.

[Beaudet, 1978] Beaudet, P. R. (1978). Rotational invariant image operators. In *Proceedings of the International Conference on Pattern Recognition*, pages 579–583.

[Beirlant et al., 2001] Beirlant, J., Dudewicz, E. J., Györfi, L., and van der Meulen, E. C. (2001). Nonparametric entropy estimation: an overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39.

[Bentley, 1975] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

[Bentley, 1990] Bentley, J. L. (1990). K-d trees for semidynamic point sets. In *proceedings of the 6th annual ACM symposium on computational geometry*, pages 187–197.

[Bertsimas and van Ryzin, 1990] Bertsimas, D. J. and van Ryzin, G. (1990). An asymptotic determination of the minimum spanning tree and minimum matching constants in geometrical probability. *Operations Research Letters*, 9:223–231.

[Bianconi and Fernández, 2007] Bianconi, F. and Fernández, A. (2007). Evaluation of the effects of gabor filter parameters on texture classification. *Pattern Recognition*, 40(12):3325–3335.

[Bigun and Granlund, 1987] Bigun, J. and Granlund, G. H. (1987). Optimal orientation detection of linear symmetry. In *proceedings of the IEEE International Conference on Computer Vision*, pages 433–438.

[Blostein and Ahuja, 1989] Blostein, D. and Ahuja, N. (1989). A multiscale region detector. *Computer Vision, Graphics, and Image Processing*, 45(1):22–41.

[Bonev et al., 2007a] Bonev, B., Escolano, F., and Cazorla, M. (2007a). A novel information theory method for filter feature selection. In *proceedings of the Mexican International Conference on Artificial Intelligence*, pages 432–440.

[Bonev et al., 2007b] Bonev, B., Escolano, F., Lozano, M. A., Suau, P., Cazrola, M., and Aguilar, W. (2007b). Constellations and the unsupervised learning of graphs. In *proceedings of the 6th IAPR Workshop on Graph Based Representations*, pages 340–250.

[Borůvka, 1926] Borůvka, O. (1926). O jistém problému minimálním. *Práca Moravské Přirodovědecké Společnosi*, 3:37–58.

[Breiman et al., 1984] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Chapman & Hall/CRC.

[Brown and Lowe, 2002] Brown, M. and Lowe, D. (2002). Invariant features from interest point groups. In *Proceedings of the 13th British Machine Vision Conference*, pages 656–665.

[Burt, 1981] Burt, P. J. (1981). Fast filter transforms for image processing. *Computer Vision, Graphics and Image Processing*, 16:20–51.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714.

[Capel and Zisserman, 1998] Capel, D. and Zisserman, A. (1998). Automated mosaicing with super-resolution zoom. In *Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition*, pages 885–891.

[Cazorla and Escolano, 2003] Cazorla, M. and Escolano, F. (2003). Two bayesian methods for junction detection. *IEEE Transactions on Image Processing*, 12(3):317–327.

[Cazorla et al., 2002] Cazorla, M., Escolano, F., Gallardo, D., and Rizo, R. (2002). Junction detection and grouping with probabilistic edge models and bayesian a*. *Pattern Recognition*, 35(9):1869–1881.

[Chen and Yuille, 2004] Chen, X. and Yuille, A. L. (2004). Detecting and reading text in natural scenes. In *proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 366–373.

[Cormen et al., 2001] Cormen, T. H., Leiserson, C. H., Rivest, R. L., and Stein, C. (2001). *Introduction to algorithms*. MIT Press, 2nd edition.

[Costa and Hero, 2004] Costa, J. A. and Hero, A. (2004). Manifold learning using euclidean k-nearest neighbor graphs. In *proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 988–991.

[Cover and Thomas, 1991] Cover, T. and Thomas, J. (1991). *Elements of information theory*. John Wiley & Sons.

[Cox, 1989] Cox, L. J. (1989). Blanche: position estimation for an autonomous robot vehicle. In *proceedings of the IEEE/RSJ international workshop on intelligent robots and systems: the autonomous robots and its applications*, pages 432–439.

[Denuit and Bellegem, 2001] Denuit, M. and Bellegem, S. V. (2001). On the stop-loss and total variation distances between random sums. *Statistics and Probability Letters*, 53:153–165.

[Deriche, 1987] Deriche, R. (1987). Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1:167–187.

[Deriche and Faugeras, 1990] Deriche, R. and Faugeras, O. D. (1990). 2-d curve matching using high curvature points: application to stereo vision. In *proceedings of the 10th International Conference on Pattern Recognition*, pages 240–242.

[Deriche and Giraudon, 1993] Deriche, R. and Giraudon, G. (1993). A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124.

[Donoser and Bischof, 2006a] Donoser, M. and Bischof, H. (2006a). 3d segmentation by maximally stable volumes (msvs). In *proceedings of the IEEE International Conference on Pattern Recognition*, volume 1, pages 63–66.

[Donoser and Bischof, 2006b] Donoser, M. and Bischof, H. (2006b). Efficient maximally stable extremal region (mser) tracking. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 553–560.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.

[Eisner, 1997] Eisner, J. (1997). State-of-the-art algorithms for minimum spanning trees: a tutorial discussion. Technical report, University of Pennsylvania.

[Eppstein et al., 1996] Eppstein, D., Italiano, G. F., Tamassia, R., Tarjan, R. E., Westbrook, J., and Yung, M. (1996). Maintenance of a minimum spanning forest in a dynamic plane graph. *Journal of algorithms*, 13(1):33–54.

[Erdogmus et al., 2004] Erdogmus, D., II, K. E. H., Príncipe, J. C., Lázaro, M., and Santamaría, I. (2004). Adaptive blind deconvolution of linear channels

using renyi's entropy with parzen window estimation. *IEEE Transactions on Signal Processing*, 52(6):1489–1498.

[Escolano et al., 2007] Escolano, F., Bonev, B., Suau, P., Aguilar, W., Frauel, Y., Sáez, J. M., and Cazorla, M. A. (2007). Contextual visual localization: cascaded submap classification, optimized saliency detection, and fast view matching. In *proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1715–1722.

[Farinella et al., 2008] Farinella, G. M., Battiato, S., Gallo, G., and Cipolla, R. (2008). Natural versus artificial scene classification by ordering discrete fourier power spectra. In *Structural, Syntactic, and Statistical Pattern Recognition (S+SSPR)*, pages 137–146.

[Fergus et al., 2003] Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271.

[Forssén, 2007] Forssén, P. E. (2007). Maximally stable colour region for recognition and matching. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[Forssén and Lowe, 2007] Forssén, P. E. and Lowe, D. G. (2007). Shape descriptors for maximally stable extremal regions. In *proceedings of the IEEE International Conference on Computer Vision*, pages 1–8.

[Fraundorfer and Bischof, 2004] Fraundorfer, F. and Bischof, H. (2004). Evaluation of local detectors on non-planar scenes. In *proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition*, pages 125–132.

[Fraundorfer and Bischof, 2005] Fraundorfer, F. and Bischof, H. (2005). A novel performance evaluation method of local detectors on non-planar scenes. In *proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, pages 33–33.

[Fraundorfer et al., 2005] Fraundorfer, F., Winter, M., and Bischof, H. (2005). Mscc: Maximally stable corner clusters. In *proceedings of the 14th Scandinavian Conference on Image Analysis*, pages 45–54.

[Frederickson, 1983] Frederickson, G. N. (1983). Data structures for on-line updating of minimum spanning trees. In *proceedings of the 15th annual ACM symposium on theory of computing*, pages 252–257.

[Fredman et al., 1986] Fredman, M. L., Sedgewick, R., Sleator, D. D., and Tarjan, R. E. (1986). The pairing heap: a new form of self-adjusting heap. *Algorithmica*, 1(1):111–129.

[Fredman and Tarjan, 1987] Fredman, M. L. and Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615.

[Friedman et al., 1975] Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1975). An algorithm for finding best matches in logarithmic expected time. Technical report, Computer Science Stanford.

[Friedman and Rafsky, 1979] Friedman, J. H. and Rafsky, L. C. (1979). Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *Annals of statistics*, 7:697–717.

[Förstner, 1986] Förstner, W. (1986). A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26:150–166.

[Gabow et al., 1986] Gabow, H. N., Galil, Z., Spencer, T., and Tarjan, R. E. (1986). Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6:109–122.

[Geman and Jdeynak, 1996] Geman, D. and Jdeynak, B. (1996). An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14.

[Gilles, 1998] Gilles, S. (1998). *Robust description and matching of images*. PhD thesis, University of Oxford.

[Goodman and O'Rourke, 2004] Goodman, J. E. and O'Rourke, J., editors (2004). *Handbook of discrete and computational geometry*. CRC Press, 2nd edition.

[Gårding and Lindeberg, 1996] Gårding, J. and Lindeberg, T. (1996). Direct computation of shape cues using scale-adapted using scale-adapted spatial derivative operators. *International Journal of Computer Vision*, 17(2):163–191.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *proceedings of the 4th ALVEY Vision Conference*, pages 147–152.

[Havrda and Charvát, 1967] Havrda, J. and Charvát, F. (1967). Quantification method of classification processes. concept of structural α-entropy. *Kybernetika*, 3:30–35.

[Henze and Penrose, 1999] Henze, N. and Penrose, M. (1999). On the multivariate runs test. *Annals of Statistics*, 27:290–298.

[Hero et al., 2003] Hero, A., Costa, J. A., and Ma, B. (2003). Asymptotic relations between minimal graphs and α-entropy. Technical report, The University of Michigan.

[Hero et al., 2002] Hero, A., Ma, B., Michel, O., and Gorman, J. (2002). Applications of entropic spanning graphs. *IEEE Signal Processing Magazine*, 19(5):85–95.

[Hero and Michel, 1999] Hero, A. and Michel, O. (1999). Asymptotic theory of greedy approximations to minimal k-point random graphs. *IEEE Transactions on Information Theory*, 45(6):1921–1939.

[Hough, 1962] Hough, P. V. C. (1962). Methods and means for recognizing complex patterns. Technical report, U. S. Patent 3.069.654.

[Kadir et al., 2003] Kadir, T., , and Brady, M. (2003). Scale saliency: a novel approach to salient feature and scale selection. In *proceedings of the 2003 International Conference on Visual Information Engineering*, pages 25–28.

[Kadir and Brady, 2001] Kadir, T. and Brady, M. (2001). Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105.

[Kadir et al., 2004] Kadir, T., Zisserman, A., and Brady, M. (2004). An affine invariant salient region detector. In *proceedings of the 8th European Conference on Computer Vision*, pages 404–416.

[Katriel et al., 2003] Katriel, I., Sanders, P., Träff, J. L., and Tra, J. L. (2003). A practical minimum spanning tree algorithm using the cycle property. In *proceedings of the 11th European Symposium on Algorithms*, pages 679–690.

[Koenderink and Richards, 1988] Koenderink, J. J. and Richards, W. (1988). Two-dimensional curvature operators. *Journal of the Optical Society of America A: Optics, Image Science and Vision*, 5(7):1136–1141.

[Konishi et al., 2003] Konishi, S., Yuille, A. L., Coughlan, J. M., and Zhu, S. C. (2003). Statistical edge detection: learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):57–74.

[Kozachenko and Leonenko, 1987] Kozachenko, L. and Leonenko, N. (1987). On statistical estimation of entropy of a random vector. *Problems of Information Transmission*, 23:95–101.

[Kruskal, 1956] Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50.

[Lazebnik et al., 2005] Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1265–1278.

[Leibe et al., 2004] Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in Computer Vision*, pages 17–32.

[Leonenko et al., 2008] Leonenko, N., Pronzato, L., and Savani, V. (2008). A class of rényi estimators for multidimensional densities. *Annals of statistics*, 36(5):2153–2182.

[Lin et al., 2005] Lin, Z., Kim, S., and Kweon, S. (2005). Robust invariant features for object recognition and mobile robot navigation. In *proceedings of the 2005 IAPR Conference on Machine Vision Applications*, pages 55–58.

[Lindeberg, 1994] Lindeberg, T. (1994). Junction detection with automatic selection of detection scales and localization scales. In *proceedings of the 1st International Conference on Image Processing*, volume 1, pages 924–928.

[Lindeberg, 1998] Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116.

[Lindeberg, 2008] Lindeberg, T. (2008). Scale-space. *Encyclopedia of Computer Science and Engineering*, 4:2495–2504.

[Lindeberg and Gårding, 1997] Lindeberg, T. and Gårding, J. (1997). Shape-adapted smoothing in estimation of 3-d shape cues from affine distortions of 2-d brightness structure. *Image and Vision Computing*, 15(6):415–434.

[Lipschutz, 1969] Lipschutz, M. M. (1969). *Differential Geometry*. McGraw Hill.

[Loupias et al., 2000] Loupias, E., Sebe, N., Bres, S., and Jolion, J. M. (2000). Wavelet-based salient points for image retrieval. In *proceedings of the 2000 IEEE International Conference on Image Processing*, volume 2, pages 518–521.

[Lowe, 1999] Lowe, D. (1999). Object recognition from local scale-invariant features. In *proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

[Lowe, 2004] Lowe, D. (2004). Distinctive image features from scale invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679.

[Mallat, 1989] Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693.

[Marr and Hildreth, 1980] Marr, D. and Hildreth, E. (1980). Theory of edge detection. In *proceedings of the Royal Society B207*, pages 187–217.

[Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767.

[Mikolajczyk et al., 2005a] Mikolajczyk, K., Leibe, B., and Schiele, B. (2005a). Local features for object class recognition. In *proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1792–1799.

[Mikolajczyk and Schmid, 2001] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *Proceedings of the Eighth IEEE International Conference on Computer Vision and Pattern Recognition*, pages 525–531.

[Mikolajczyk and Schmid, 2002] Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision*, pages 128–142.

[Mikolajczyk and Schmid, 2004a] Mikolajczyk, K. and Schmid, C. (2004a). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

[Mikolajczyk and Schmid, 2004b] Mikolajczyk, K. and Schmid, C. (2004b). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

[Mikolajczyk et al., 2005b] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005b). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72.

[Moravec, 1977] Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. In *proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 584–584.

[Moret and Shapiro, 1991] Moret, B. M. E. and Shapiro, H. D. (1991). An empirical analysis of algorithms for constructing a minimum spanning tree. In *proceedings of the 2nd workshop on algorithms and data structures*, pages 400–411.

[Murphy-Chutorian and Trivedi, 2006] Murphy-Chutorian, E. and Trivedi, M. (2006). N-tree disjoint-set forests for maximally stable extremal regions. In *proceedings of the 17th British Machine Vision Conference*.

[Nagel, 1983] Nagel, H. H. (1983). Displacement vectors derived from second-order intensity variations in image sequences. *Graphical Models and Image Processing*, 21:85–117.

[Neemuchwala, 2004] Neemuchwala, H. (2004). *Entropic graphs for image registration*. PhD thesis, University of Michigan.

[Neemuchwala et al., 2006] Neemuchwala, H., Hero, A., Zabuawala, S., and Carson, P. (2006). Image registration methods in high-dimensional space. *International Journal of Imaging Systems and Technology*, 16(5):130–145.

[Newman et al., 2006] Newman, P., Cole, D., and Ho, K. (2006). Outdoor slam using visual appearance and laser ranging. In *proceedings of the IEEE International Conference on Robotics and Automation*, pages 1180–1187.

[Noble, 1988] Noble, J. A. (1988). Finding corners. *Image and Vision Computing*, 6(2):121–128.

[Pettie and Ramachandran, 2002] Pettie, S. and Ramachandran, V. (2002). An optimal minimum spanning tree algorithm. *Journal of the ACM*, 49(1):16–34.

[Peñalver et al., 2006] Peñalver, A., Escolano, F., and Sáez, J. M. (2006). Two entropy-based methods for learning unsupervised gaussian mixture models. In *proceedings of the 6th International Workshop on Statistical Pattern Recognition*, pages 649–657.

[Peñalver et al., 2009] Peñalver, A., Escolano, F., and Sáez, J. M. (2009). Learning gaussian mixture models with entropy-based criteria. *IEEE Transactions on Neural Networks*, 20(11):1756–1771.

[Prim, 1957] Prim, R. C. (1957). Shortest connection networks and some generalization. *Bell System Technical Journal*, 36:1389–1401.

[Rényi, 1961] Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561.

[Rohr, 1990] Rohr, K. (1990). Uber die modellierung und identifikation charakteristischer grauwertverlaufe in realweltbildern. In *DAGM Symposium Mustererkennung*, pages 24–26.

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *proceedings the 9th European Conference on Computer Vision*, pages 430–443.

[Rubner et al., 2000] Rubner, Y., Tomasi, C., and Guibas, L. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

[Schmid and Mohr, 1997] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534.

[Schmid et al., 2000] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172.

[Sebe and Lew, 2001] Sebe, N. and Lew, M. S. (2001). Comparing salient point detectors. In *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo*, pages 64–67.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: a text retrieval approach to object matching in videos. In *proceedings of the Ninth International Conference on Computer Vision*, volume 2, pages 1470–1470.

[Smith and Brady, 1995] Smith, S. M. and Brady, M. (1995). Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

[Sobel and Feldman, 1968] Sobel, I. and Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *presented at a talk at the Stanford Artificial Intelligence Project*.

[Stasko, 1987] Stasko, J. T. (1987). Pairing heaps: experiments and analysis. *Communications of the ACM*, 30(3):243–249.

[Stollnitz et al., 1995] Stollnitz, E., DeRose, T., and Salesin, D. (1995). Wavelets for computer graphics: a primer, part 1. *IEEE Computer Graphics and Applications*, 15(3):76–84.

[Stowell and Plumbley, 2009] Stowell, D. and Plumbley, M. D. (2009). Fast multidimensional entropy estimation by k-d partitioning. *IEEE Signal Processing Letters*, 16(6):537–540.

[Sáez and Escolano, 2006] Sáez, J. M. and Escolano, F. (2006). 6dof entropy minimization slam. In *proceedings of the IEEE International Conference on Robotics and Automation*, pages 1548–1555.

[Thrun et al., 2001] Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.

[Tomasi and Kanade, 1991] Tomasi, C. and Kanade, T. (1991). Detection and tracking of point features. Technical report, Carnegie Mellon University.

[Torralba and Oliva, 2003] Torralba, A. and Oliva, A. (2003). Statistics of natural image categories. In *Network: Computation in Neural Systems*, pages 391–412.

[Trajkovic and Hedley, 1998] Trajkovic, M. and Hedley, M. (1998). Fast corner detection. *Image and Vision Computing*, 16(2):75–87.

[Tsallis, 1988] Tsallis, C. (1988). Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52:479–487.

[Tuytelaars and Gool, 2004] Tuytelaars, T. and Gool, L. V. (2004). Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85.

[Vedaldi, 2007] Vedaldi, A. (2007). An implementation of multi-dimensional maximally stable extremal regions. Technical report, University of California.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Fast and robust classification using asymmetric adaboost and a detector cascade. In *Advances in Neural Information Processing Systems*, pages 1311–1318.

[Vosselman and Dijkman, 2001] Vosselman, G. and Dijkman, S. (2001). 3d building model reconstruction from point clouds and ground plans. *International Archives of the Photogrammetry*, 34(3):22–24.

[Wang and Brady, 1994] Wang, H. and Brady, M. (1994). A practical solution to corner detection. In *Proceedings of the 1994 International Conference on Image Processing*, volume 1, pages 919–923.

[Witkin, 1983] Witkin, A. P. (1983). Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 1019–1022.

[Yuille and Poggio, 1986] Yuille, A. L. and Poggio, T. A. (1986). Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25.

[Zhao et al., 1990] Zhao, L. C., Krishnaiah, P. R., and Chen, X. R. (1990). Almost sure l_r-norm convergence for data based histogram density estimates. *Theory of Probability and its Applications*, 35(2):396–403.

Glossary Index